TAMPEREEN TEKNILLINEN YLIOPISTO

RISTO MÄNTYLÄ

TALLENNUSVERKON HALLINNAN JA MONITOROINNIN KEHITTÄMINEN LAAJASSA TUOTANTOYMPÄRISTÖSSÄ

Diplomityö

Tarkastaja: professori Jarmo Harju

Tarkastaja ja aihe hyväksytty

Tieto- ja sähkötekniikan

tiedekuntaneuvoston

kokouksessa 13.1.2010

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Signaalinkäsittelyn ja tietoliikennetekniikan koulutusohjelma

MÄNTYLÄ, RISTO: Tallennusverkon hallinnan ja monitoroinnin kehittäminen laajassa tuotantoympäristössä

Diplomityö, 92 sivua, 10 liitesivua

Huhtikuu, 2010

Pääaine: Tietoliikenneverkot ja protokollat

Työn tarkastajat: professori Jarmo Harju ja R&D Specialist Jussi Hiltunen

Avainsanat: tallennusverkko, SAN, Fibre Channel, SCSI, HP Storage Essentials SRM

Tämän diplomityön tarkoitus on ollut tutustua tallennusverkkojen toimintaan ja kehittää tallennusverkon hallintaa ja monitorointia erään yrityksen datakeskuksessa. Työssä on otettu käyttöön HP Storage Essentials SRM -hallintasovellus ja kehitetty sen ominaisuuksia kyseisen yrityksen tarpeita varten.

Tallennusverkolla (SAN) tarkoitetaan joukkoa tietokoneita ja tallennuslaitteita, jotka on yhdistetty toisiinsa nopean tietoliikenneverkon kautta. Tallennusverkon tehtävä on tiedon tallentaminen, suojaaminen ja hallinta. Tallennusverkko voidaan toteuttaa monella eri teknisellä ratkaisulla. Suosituin tallennusverkkotekniikka tällä hetkellä on Fibre Channel -protokollaan perustuva tallennusverkko. Tässä diplomityössä keskitytään tallennusverkon toteuttamiseen SCSI ja Fibre Channel -protokollien avulla.

Diplomityön kohteena oleva datakeskus sisältää yhden Suomen suurimmista tallennusverkoista. Tallennusverkon koko on kasvanut viime vuosina erittäin paljon, minkä seurauksena tarve keskitetylle hallinta- ja monitorointijärjestelmälle on syntynyt. Hallintaohjelmistoksi tallennusverkkoon on valittu Hewlett-Packardin valmistama HP Storage Essentials SRM Enterprise Edition. Storage Essentialsin lisäksi hallintaan käytetään myös muita sovelluksia, jotka keräävät tietoa SE:n puolesta.

Diplomityön tuloksia ovat tallennusverkon keskitetty hallinta ja monitorointi sekä Storage Essentialin agenttien automaattinen jakelujärjestelmä. Keskitetty hallinta koostuu useasta sovelluksesta, jotka on sijoitettu kolmelle eri palvelimelle. Agenttien automaattinen jakelujärjestelmä huolehtii siitä, että agentit ovat asennettuina hallittavilla palvelimilla aina, vaikka palvelimen käyttöjärjestelmä asennettaisiin uudelleen joka päivä.

Storage Essentials soveltuu hyvin tallennusverkon tiedon keräämiseen ja jäsentämiseen, mutta laitteiden konfigurointi on toteutettu toistaiseksi varsin puuttellisesti. Sovelluksen toinen ongelma on sen porttiperusteinen lisensointi.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Signal Processing and Communications Engineering

MÄNTYLÄ, RISTO: The Development of Storage Area Network's Management and Monitoring in a Large Production Environment

Master of Science Thesis, 92 pages, 10 enclosure pages

April, 2010

Major: Communication Networks and Protocols

Examiners: Professor Jarmo Harju and R&D Specialist Jussi Hiltunen

Keywords: SAN, Fibre Channel, SCSI, HP Storage Essentials SRM

The goal for this Master of Science Thesis is to get familiar with storage area networks (SANs), to develop management and monitoring of a storage area network by taking into use HP Storage Essentials SRM software on the datacenter and to further develop the features provided by the Storage Essentials.

Storage Area Network is a group of computers and storage devices connected to each other through a fast and reliable communications network. The purpose of a SAN is to save, protect and manage information. A SAN can be implemented with a number of different technologies. The most popular SAN technology is Fibre Channel protocol. This Master of Science Thesis concentrates on SAN based on Fibre Channel.

The datacenter that is the target of this thesis has one of the largest SANs in Finland. The size of the SAN has grown rapidly during past few years. This has been the driving force to implement a centralized management and monitoring for the SAN. HP Storage Essentials SRM Enterprise Edition has been selected to be the centralized management console. Storage Essentials cannot gather all the information by itself, so also other softwares are required.

The results of this thesis are centralized management and monitoring of the SAN and an automatic distribution system for the Storage Essentials agent software. Centrilized management consists of several pieces of software that have been installed on three different servers. The automatic agent distribution system is responsible that every server has the agent software installed always even if the server's operating system is reinstalled every day.

Storage Essentials is a very efficient tool to gather and analyze information from the SAN but it still lacks a good tool to configure devices in the SAN. Another bad feature of the Storage Essentials is the licensing policy that is based on the amount of ports in the SAN.

ALKUSANAT

Tämä diplomityö on tehty eräälle suomalaislähtöiselle monikansalliselle yritykselle, jonka nimeä ei diplomityössä mainita siitä syystä, että diplomityöstä voitiin tehdä julkinen. Pääaineopintojeni ansiosta olen saanut vankan kokemuksen perinteisten tietoliikenneverkkojen toiminnasta. Tämän diplomityön ansiosta olen saanut erittäin arvokasta kokemusta myös täysin erilaisen verkon toiminnasta. Haluan kiittää kyseistä yritystä erittäin mielenkiintoisesta ja opettavaisesta diplomityöaiheesta, joka osoittautui työn tekemisen edetessä yhä kiinnostavammaksi. Toivon, että tulevaisuudessa saan työskennellä lisää tallennusverkkojen parissa.

Esitän kiitokseni työni tarkastajalle professori Jarmo Harjulle sekä työni ohjaajalle R&D Specialist Jussi Hiltuselle. Haluan myös kiittää Simo Syväjärveä, jolta sain arvokkaita ohjeita tallennusverkkojen toiminnasta, sekä vaimoani Sallaa, joka on kannustanut minua erittäin paljon diplomityön teon jokaisessa vaiheessa.

Tampereella 16.4.2010

_____

Risto Mäntylä

SISÄLLYSLUETTELO

LYHENTEET

ANSI	American National Standards Institute on voittoa tavoittelematon standardointiorganisaatio.
BER	Bit Error Rate kuvaa tiedonsiirrossa tapahtuvien virheiden määrää kaikkia siirrettyjä bittejä kohden.
CE	Core-Edge on tallennusverkoissa käytetty verkkotopologia.
CIM	Common Information Model on avoin standardi tietojärjestelmän elementtien hallintaan.
CIMOM	CIM Object Manager on CIM:n komponentti, joka tiedot hallittavista komponenteista.
CMS	Central Management Server on SIM:n komponentti, joka vastaa datakeskuksen keskitetystä hallinnasta.
CRC	Cyclic Redundancy Check on virheentarkistusmekanismi, jonka avulla voidaan havaita tiedonsiirrossa tapahtuvia virheitä.
CV	HP Command View EVA sovellus, jolla voidaan hallita keskitetysti EVA-levyjärjestelmiä.
CV-TL	HP Command View for Tape Libraries on sovellus, jolla voidaan hallita HP:n nauhakirjastoja.
DAS	Direct Attached Storage on suoraan palvelimeen kiinnitetty tai palvelimen sisällä oleva massamuistilaite.
DMTF	Distributed Management Task Force on standardointiorganisaatio, joka pyrkii kehittämään tietojärjestelmien hallintaa.
DNS	Domain Name System on hierarkinen nimipalvelu, joka yhdistää IP-osoitteet helpommin muistettaviin nimiin.
DWDM	Dense Wavelength Division Multiplexing on optinen kanavointimenetelmä, jota käytetään tiedon siirrossa pitkillä välimatkoilla.
EOF	End of Frame on protokollakehyksen lopetusmerkki.
EVA	Enterprise Virtual Array on HP:n valmistama levyjärjestelmä.
FATA	Fibre Channel ATA kiintolevy.
FC	Fibre Channel on nopea verkkoprotokolla, jota käytetään pääasiassa tallennusverkoissa.
FC-AL	Fibre Channel Arbitrated Loop on FC-standardi, joka määrittelee rengas-topologian käytön FC:ssä.
FC-GS	Fibre Channel Generic Services määrittelee Fibre Channelin hallintapalvelut.

FCIA	Fibre Channel Industry Association on voittoa tavoittelematon standardointiorganisaatio, joka kehittää Fibre Channeliin liittyviä teknologioita
FCIP	Fibre Channel over IP on verkkoprotokolla, jolla voidaan kuljettaa FC-kehyksiä IP-verkon ylitse yhdistäen erillisiä FC-saarekkeita.
FC-MI	Fibre Channel Methodologies for Interconnects määrittelee perusoperaatiot, jotka takaavat eri laitteiden välisen yhteensopivuuden.
FCoCEE	Fibre Channel over Convergence Enhanced Ethernet on verkkoteknologia, jolla pyritään parantamaan tavallista Ethernet-lähiverkko tallennusverkon käyttöön sopivaksi
FCoE	Fibre Channel over Ethernet on tallennusverkon protokolla, jossa FC-protokollaa kuljetetaan Ethernet-lähiverkossa
FCP	Fibre Channel Protocol määrittelee miten SCSI-komennot sovitetaan FC-siirtokanavaan.
FCZS	Fibre Channel Zone Server on tallennusverkon kytkimen tarjoama palvelu vyöhykkeiden määrittelemiseksi tallennusverkkoon.
FOS	Fabric OS on Brocaden tallennusverkon kytkimissä oleva käyttöjärjestelmä.
FSPF	Fabric Shortest Path First on tallennusverkoissa käytettävä reititysprotokolla.
GBIC	Gigabit Interface Converter on lähetin-vastaanotin, joka muuntaa sähköisen signaalin optiseksi ja toisin päin.
HA	High Availability on IT-järjestelmissä käytetty termi, joka takaa korkeaa saatavuutta.
HBA	Host Bus Adapter on palvelimien laajennuskortti, joka sisältää liitynnän esimerksi Fibre Channel-tallennusverkkoon.
HCA	Host Channel Adapter on Infiniband-tekniikassa palvelimen kanavasovitin, jolla palvelin liittyy InfiniBand-verkkoon.
HP-UX	HP-UX on Hewlett-Packardin versio UNIX-käyttöjärjestelmästä.
HTTP	Hyper Text Transfer Protocol on tiedonsiirtoprotokolla, jota käytetään yleisesti esimerksi web-sivujen siirtämiseen palvelimen ja selaimen välillä.
HTTPS	HTTP Secure on turvallinen versio HTTP:stä, jossa tieto salataan tiedonsiirron ajaksi.
I/O	Input/Output on termi, joka tarkoittaa tiedonsiirtoa kahden järjestelmän välillä.
IBTA	InfiniBand Trade Association on standardointiorganisaatio, joka vastaa InfiniBand-tekniikan kehittämisestä.
IEEE	Institute of Electrical and Electronics Engineers on voittoa tavoittelematon standardointiorganisaatio.

IETF	Internet Engineering Task Force määrittelee ja kehittää Internetiin liittyviä standardeja.
iFCP	Internet Fibre Channel Protocol on yhdyskäytäväprotokolla, jonka avulla voidaan muodostaa FC-laitteille tallennusverkko käyttämällä TCP/IP-verkkoa
IFL	Inter Fabric Link on fyysinen linkki, joka yhdistää kaksi kudosta toisiinsa.
IP	Internet Protocol on yleisin verkkokerroksen protokolla.
iSCSI	Internet SCSI on tallennusverkkoprotokolla, jossa SCSI-komentoja kuljetataan suoraan IP-protokollan päällä.
ISL	Inter Switch Link on fyysinen linkki, joka yhdistää kaksi kytkintä toisiinsa.
JBOD	Just a Bunch Of Disks kuvaa kokoelmaa kiintolevyjä, jotka toimivat kaikki itsenäisesti.
LAN	Local Area Network, lähiverkko.
LDEV	Logical Device on levyjärjestelmissä käytetty nimitys loogisesta kiintolevystä.
LUN	Logical Unit Number on SCSI-protokollassa käytettävä nimeämiskäytäntö, jolla identifioidaan levyjä tai levyosioita.
MAN	Metropolitan Area Network on laaja tietokoneverkko, joka käsittää esimerkiksi koko kaupungin
MIB	Management Information Base on SNMP:ssä käytettävä hierarkinen tietokanta hallittavien objektien määrittelemiseen.
NAS	Network Attached Storage on verkkolevyjärjestelmä, jossa tiedonvälitys tapahtuu tiedostojärjestelmätasolla.
NMS	Network Management System on SNMP-palvelin, joka kerää muiden laitteiden lähettämiä SNMP-viestejä.
OSI-malli	Open Systems Interconnection Reference Model on abtrakti kuvaus tietoliikenneverkkoprotokollien kerrosmallista.
PCI	Peripheral Component Interconnect on tietokoneissa käytettävä väyläarkkitehtuuri.
QAS	Quick Arbitration and Selection on SCSI-väylän varaamiseen käytettävä protokolla.
RAID	Redundant Array of Independent Disks on tekniikka, jolla voidaan parantaa kiintolevyjen suorituskykyä tai luotettavuutta.
RHEL	Red Hat Enterprise Linux on suosittu linux-distribuutio.
RO	HP Report Optimizer SRM on Storage Essentialin lisäsovellus, jolla voidaan koostaa raportteja tietokannasta.
SAN	Storage Area Network eli tallennusverkko.
SATA	Serial ATA on kiintolevyjen liittämiseen käytetty sarjamuotoinen väylätekniikka.

SCP	Secure Copy on tiedostonsiirtoprokolla, joka perustuu SSH:n käyttöön.
SCSI	Small Computer System Interface on yleisesti palvelimien I/O-väylässä käytettävä tiedonsiirtoprotokolla.
SE	HP Storage Essentials SRM on keskitetty hallintasovellus tallennusverkkojen hallintaan.
SFTP	Secure File Transfer Protocol on tiedostonsiirtoprotokolla, joka perustuu SSH:n käyttöön.
SIM	HP Systems Insight Manager on hallintasovellus, jolla voidaan hallita monia erilaisia datakeskuksen laitteita.
SMI-A	Brocade SMI Agent on SMI-S välityspalvelinsovellus, jonka kautta voidaan hallita tallennusverkon kytkimiä.
SMI-S	Storage Management Initiative - Specification on WBEM:n ja CIM:n laajennus tallennusverkkojen hallintaan.
SNIA	Storage Networking Industry Association on standardointiorganisaatio, joka kehittää tallennusverkkoon liittyviä standardeja.
SNMP	Simple Network Management Protocol on yleisesti käytetty verkonhallintaprotokolla.
SOF	Start of Frame on protokollakehyksen aloitusmerkki.
SPI	SCSI Parallel Interface määrittelee SCSI:n ketjuväylän toiminnan.
SSD	Solid State Drive on kiintolevytekniikka, jossa käytetään magneettisten levyjen sijasta flash-muistia tiedon varastointiin.
SSH	Secure Shell on protokolla, jonka avulla voidaan muodostaa salattu etäyhteys palvelimen komentotulkkiin.
SUSE	SUSE on eräs linux-distribuutio.
TCA	Target Channel Adapter on InfiniBand-tekniikassa levyjärjestelmän kanavasovitin, jolla levyjärjestelmä liittyy InfiniBand-verkkoon.
TCP/IP	Transmission Control Protocol / Internet Protocol on protokollaperhe, jota käytetään Internetissä tiedonsiirtoon.
THP	Thin Provisioning on tekniikka, jota käytetään levyjärjestelmissä levytilan tehokkaaseen hallintaan.
VLAN	Virtual LAN on tekniikka, jolla voidaan jakaa lähiverkko useisiin virtuaalisiin lähiverkkoihin.
WAN	Wide Area Network on tietokoneverkko, joka kattaa laajan maantieteellisen alueen.
WBEM	Web-Based Enterprise Management on hallintaprotokolla, jota käytetään hajautettujen järjestelmien hallintaan.
WWN	World Wide Name on FC-tallennusverkossa käytettävä osoite.
XML	Extensible Markup Language on kuvauskieli, jolla voidaan kuvata rakenteisia dokumentteja.

1 JOHDANTO

Tallennusverkolla (SAN) tarkoitetaan joukkoa tietokoneita ja tallennuslaitteita (storage device), jotka ovat yhdistetty toisiinsa nopean tietoliikenneverkon kautta. Tallennusverkko eroaa verkkokiintolevystä (NAS) siinä, että tallennusverkoissa data siirretään lohkotasolla ja verkkokiintolevyissä tiedostojärjestelmätasolla. Tallennusverkot toimivat siis alemmalla tasolla kuin verkkokiintolevyt. Alemmalla tasolla toimiminen vähentää tiedonsiirrosta aiheutuvaa yleisrasitetta (overhead) ja mahdollistaa minkä tahansa tiedostojärjestelmän käytön. Tallennusverkon tehtävä on tiedon tallentaminen, suojaaminen ja hallinta. Tallennusverkko voidaan toteuttaa monella eri teknisellä ratkaisulla. Suosituin tallennusverkkotekniikka tällä hetkellä on Fibre Channel -protokollaan perustuva tallennusverkko. Sen markkinaosuus on tämän diplomityön kirjoitushetkellä noin 80 prosenttia. Tallennusverkko voidaan toteuttaa myös Ethernet-tekniikkaan ja IP-protokollaan perustuvilla ratkaisulla. Tässä diplomityössä keskitytään tallennusverkon toteuttamiseen SCSI- ja Fibre Channel-protokollan avulla.

Tallennusverkoista muodostuu nopeasti erittäin laajoja ja vaikeasti hallittavia kokonaisuuksia. Tästä syystä tallennusverkon hallinta pyritään keskittämään yhteen pisteeseen, josta verkon laitteita voidaan konfiguroida ja monitoroida. Tallennusverkon hallinta eroaa perinteisestä verkonhallinnasta siten, että tallennusverkossa myös datan turvallinen säilytys kuuluu hallinnan työtehtäviin. Perinteisessä verkonhallinnassa tavoite on ainoastaan taata turvallinen datan kuljetus verkkoinfrastruktuurin läpi. Se mitä datalle tapahtuu sen ohitettua kyseisen verkkoinfrastruktuurin, ei koske perinteistä verkonhallintaa. Tallennusverkossa sekä datan turvallinen kuljetus että säilöntä ovat päätavoitteita.

Tämän diplomityön päätavoite on kehittää yhden Suomen suurimman tallennusverkon hallintaa ja monitorointia. Hallinnan keskipisteenä on tarkoitus käyttää HP Storage Essentials SRM -sovellusta, joka on tarkoitettu heterogeenisen tallennusverkon hallintaan. Tallennusverkko koostuu noin 900 fyysisestä ja 700 virtuaalisesta palvelimesta sekä useista levyjärjestelmistä. Työn muita tavoitteita ovat tallennusverkon toimintaan tutustuminen sekä Storage Essentials -sovelluksen toimintojen jatkokehittäminen kyseistä ympäristöä varten.

Diplomityö koostuu viidestä luvusta, joista ensimmäinen on johdanto. Toisessa luvussa perehdytään syvällisesti Fibre Channelin avulla toteutetun tallennusverkon toimintaan. Luvussa käydään läpi tallennusverkon edut, rajoitteet, tavallisimmat laitteet, verkon erilaiset topologiat sekä hallintaan käytettävät protokollat. Luvussa esitellään myös lyhyesti muita tekniikoita, joiden avulla tallennusverkko voidaan toteuttaa. Kolmannessa luvussa on dokumentoituna diplomityön kohteena oleva tallennusverkko.

Luvussa käydään läpi kaikki kyseisen tallennusverkon hallintaan käytettävät ohjelmistot, hallintapalvelimet, tallennusverkossa olevat laitteet sekä verkon topologia. Neljäs luku sisältää diplomityön tulokset ja viides luku johtopäätökset.

2 TALLENNUSVERKOT

Tallennusverkko uudistaa merkittävästi datakeskuksen rakennetta. Tietokoneiden aikakauden alussa kaikki laskenta oli keskitetty yhdelle suurtietokoneelle (mainframe), joka pystyi ajamaan kaikkia organisaation sovelluksia. Suurtietokoneen koko laitteisto oli koottuna yhteen paikkaan, datakeskukseen. Kaikki sen tarvitsema tallennustila oli sijoitettuna yhdessä paikassa, joten sitä oli helppo hallita.

PC-tietokoneiden vallankumouksen myötä laskentateho ja tallennustila alkoivat hajaantua ympäri organisaatiota useille yksittäisille tietokoneille. Tällaiset tietokoneet, palvelimet, sijoitettiin lähelle niiden käyttäjiä. Palvelimet yhdistettiin lähiverkoiksi LAN-tekniikan avulla ja myöhemmin Internetin myötä tietoa pystyttiin siirtämään yhä pidempiä etäisyyksiä. Palvelimet pystyivät jakamaan tiedostoja keskenään lähiverkon ja Internetin kautta, mutta fyysisen levytilan jakaminen muiden palvelimien kanssa oli mahdotonta. Palvelimien hajanainen sijoittaminen muodostui ongelmaksi, koska kaikki tieto oli palvelimien sisällä ympäri organisaatiota. Hajallaan olevan infrastruktuurin hallinta muodostui todella vaikeaksi. Sovellukset, joilla on korkeat suoritusvaatimukset, tarvitsevat yhteyden fyysiseen tallennustilaan lohkotasolla (block level). Vaikka lähiverkko ja TCP/IP tarjoaisivat mahdollisuuden lohkotason pääsyyn etäällä oleville kiintolevyille, ne eivät ole suunniteltu sovelluksille, joilla on korkeat vaatimukset esimerkiksi viiveelle. Nämä asiat ovat johtaneet tallennusverkkojen kehittämiseen. Tallennusverkon avulla voidaan yhdistää suurtietokone- ja hajautetun palvelin-arkkitehtuurin hyvät puolet.

Tallennusverkko eroaa muista tallennusratkaisuista siten, että tietokoneiden tai palvelimien tallennustila on siirretty verkossa sijaitseville tallennuslaitteelle. Tallennusverkon avulla massamuistilaitteiden tallennustila voidaan jakaa monien palvelimien kesken tehokkaasti. Viimeaikoina voimakkaasti yleistyneet pilvipalvelut, ovat varmasti nostaneet tallennusverkkoihin kohdistunutta mielenkiintoa voimakkaasti.

2.1 Datakeskusten arkkitehtuurit

Datakeskuksen arkkitehtuuri on joko palvelinkeskeinen tai levyjärjestelmäkeskeinen riippuen siitä, käytetäänkö datakeskuksessa tallennusverkkoa vai ei.

2.1.1 Palvelinkeskeinen arkkitehtuuri

Palvelinkeskeisellä arkkitehtuurilla (server-centric architecture) tarkoitetaan perinteistä datakeskuksen rakennetta, jossa datan varastointiin käytetty fyysinen levytila on sisällä

palvelimissa tai niihin suoraan kytketyissä laitteissa. Palvelimet ovat keskeisessä roolissa tiedon varastoinnissa ja käsittelyssä.

Palvelinkeskeisen arkkitehtuurin ongelmaksi muodostuu nopeasti levytilan epätasainen jakautuminen palvelimien kesken. Palvelimet, joissa levytila on vähissä, on vaikea ottaa käyttöön levytilaa, joka on mahdollisesti vapaana muissa palvelimissa. Muiden palvelimien dataan ei myöskään pääse käsiksi ilman, että palvelimet ovat yhteydessä toisiinsa esimerkiksi lähiverkon kautta. Palvelimen rikkoutuminen aiheuttaa sen, että palvelimen dataan ei pääse käsiksi ilman fyysisten laitteiden siirtoa.

Vaikka kiintolevyjen ja nauhojen tallennustiheys kasvaa jatkuvasti, tarvittavan tallennustilan määrä kasvaa vielä nopeammin. Paikallisten kiintolevyjen ja I/O-porttien määrä palvelimissa on rajallinen. Lisäksi SCSI-kaapelit eivät voi olla pidempiä kuin 25 metriä. Tästä johtuen perinteisillä keinoilla palvelimeen lisättävä tallennustila on erittäin rajallista.

Palvelinkeskeinen arkkitehtuuri ei myöskään sovellu palvelinklusterien käyttöön, koska klusterin palvelimilla on ainakin osittain yhteinen tallennustila, jota kaikki voivat käyttää samanaikaisesti. Palvelinkeskeisessä arkkitehtuurissa ei voida käyttää tehokkaita varmuuskopiointimenetelmiä, koska muut järjestelmät eivät pääse suoraan käsiksi palvelimen kiintolevyihin, vaan niiden on käytettävä esimerkiksi lähiverkkoyhteyttä palvelimen datan varmuuskopiointiin. Tämä aiheuttaa ylimääräistä kuormaa palvelimen verkkoyhteyksiin sekä suorituskapasiteettiin. Varmuuskopiointiin voidaan käyttää omaa lähiverkkoinfrastruktuuria, mutta tämä lisää merkittävästi laitekustannuksia eikä poista muuta palvelimeen aiheutuvaa kuormaa.

Näistä syistä perinteinen palvelinkeskeinen arkkitehtuuri ei ole enää riittävä laajoissa datakeskuksissa. Tallennusverkot ja suuret massamuistilaitteet keskittävät palvelimien levytilan tarpeen helpommin hallittaviksi kokonaisuuksiksi. Tallennusverkot siirtävät tiedon varastoinnin pois palvelimilta. Palvelimet eivät ole enää niin keskeisessä roolissa, koska toiminnan kannalta tärkeä tieto on siirretty tallennusverkkoon. Järjestelmästä käytetään nimitystä levyjärjestelmäkeskeinen arkkitehtuuri (storage-centric architecture).

2.1.2 Levyjärjestelmäkeskeinen arkkitehtuuri

Perinteisen palvelinkeskeisen arkkitehtuurin rajoitteet ovat johtaneet tallennusverkkojen ja keskitettyjen levyjärjestelmien kehittämiseen. Tallennusverkon ajatus on korvata SCSI-kaapelit palvelimen ja tallennuslaitteen välillä uudella erillisellä verkolla, jota käytetään ainoastaan datan välitykseen palvelimen ja tallennuslaitteen välillä.

Palvelinkeskeisessä arkkitehtuurissa palvelimet eivät pysty käyttämään toistensa kiintolevyjä muuta kuin toisen palvelimen kautta. Levyjärjestelmäkeskeisessä arkkitehtuurissa tallennuslaitteet ovat itsenäisiä laitteita, jotka sijaitsevat

tallennusverkossa. Kaikki tallennusverkkoon kytketyt palvelimet pystyvät samanaikaisesti käyttämään keskitetyn tallennuslaitteen tallennustilaa ilman muiden palvelimien apua.

Toiminnan kannalta tärkeän tiedon siirtyminen pois palvelimilta siirtää palvelimet toissijaiseen rooliin datakeskuksessa. Koska tieto sijaitsee tallennusverkossa keskitetyssä tallennuslaitteessa, datakeskuksen arkkitehtuurin painopiste siirtyy kohti tallennusverkkoa. Palvelimien rooliksi jää ainoastaan tallennuslaitteeseen tallennettavan tiedon prosessointi. Palvelimille jätetään yleensä jonkin verran paikallista tallennustilaa esimerkiksi käyttöjärjestelmää varten, mutta käyttöjärjestelmä voidaan ladata myös tallennusverkosta, jolloin palvelimen rooli datakeskuksessa pienenee entisestään.

Tallennusverkkojen käyttö vähentää palvelimen sisäisten kiintolevyjen tarvetta ja vähentää huomattavasti ylläpidolle aiheutuvaa työmäärää ja kustannuksia. Kiintolevyt ovat mekaanisia laitteita, jotka ovat yleensä melko herkkiä hajoamaan. Suurissa datakeskuksissa yksittäisten kiintolevyjen määrä nousee nopeasti tuhansiin, jos tallennustilaa ei ole keskitetty tallennusverkkoon. Jos palvelimet on vielä sijoitettu moneen eri kohteeseen, aiheuttaa tallennustilan hallinta merkittäviä ylläpitokustannuksia.

Tallennusverkot mahdollistavat monia uusia menetelmiä tiedonhallintaan, jotka eivät ole mahdollisia palvelinkeskeisessä arkkitehtuurissa. Tällaisia ominaisuuksia ovat keskitetty levytilan hallinta ja tehokas jakaminen palvelimien kesken, tiedon parempi saatavuus, tehokkaat varmuuskopiointimenetelmät sekä palvelinklusterien käyttö.

Tallennusverkkojen suosio on kasvanut valtavasti viime aikoina. Suuret yritykset ovat käyttäneet jo jonkin aikaa tallennusverkkoja, mutta verkkoon tarvittavien laitteiden kysynnän kasvaessa niiden hinnat ovat tulleet myös alas. Tämä on mahdollistanut tallennusverkkojen laajenemisen myös julkishallinnon ja keskisuurien yritysten datakeskuksiin.

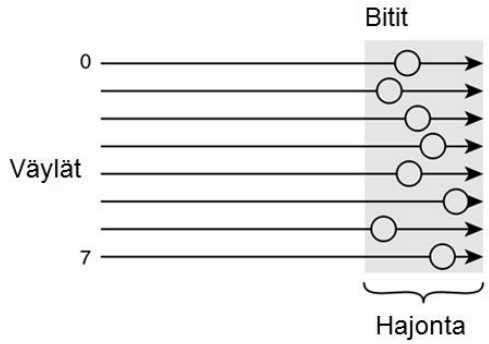
2.2 Tallennusverkon edut

Tässä luvussa käsitellään tarkemmin edellä mainittuja tallennusverkoista saatavia hyötyjä sekä rajoitteita, joita sen käytössä ilmenee.

2.2.1 Fibre Channel

SCSI-kaapelien käyttö palvelimien ja massamuistilaitteiden yhdistämisessä sisältää paljon rajoitteita, jotka johtuvat SCSI-väylän rakenteesta. Alkuperäisessä SCSI-väylässä on kahdeksan rinnakkaista dataväylää, jotka siirtävät yhden tavun bitit samanaikaisesti jokaisella kellojaksolla. Rinnakkaisrakenteella saadaan aikaan suhteellisen korkea kaistanleveys, mutta väylien rinnakkaisuus aiheuttaa myös ongelmia. Yksi ongelma on rinnakkaisrakenteisissa väylissä esiintyvä hajontailmiö (skew). Vaikka tavun bitit

lähetetään samanaikaisesti dataväylille, saattaa lähetyksessä esiintyä pieniä viive-eroja väylien välillä. Tästä johtuen bitit saapuvat eri aikaan väylän toiseen päähän. SCSI-väylässä pitää käyttää vastaanottoikkunaa, jonka aikana saapuneet bitit oletetaan kuuluvan samaan tavuun (kuva 2.1). SCSI-kaapeleissa käytetään yleensä kuparia. Rinnakkaiset kuparijohtimet aiheuttavat toisiinsa sähkömagneettisia häiriöitä, jotka heikentävät signaalia. Näistä syistä SCSI-kaapelien maksimipituus on noin 25 metriä, mikä rajoittaa massamuistilaitteiden ja palvelimien sijoittamista merkittävästi. [2, s. 32]



Kuva 2.1. Bittien hajonta SCSI-väylässä [2, s. 33]

Tallennusverkoissa käytetään yleensä kuparisten SCSI-kaapelien sijasta valokuitukaapeleita. Valokuidussa tieto siirretään laservalon avulla, joka kokonaisheijastuu kaapelin sisäpinnoista. Tiedonsiirto valokuidussa tapahtuu sarjassa, ei rinnakkaisesti kuten SCSI-väylässä. Valokuitukaapelien pituusrajat ovat paljon suuremmat kuin SCSI-kaapelien. Ne voivat olla jopa useita kymmeniä kilometrejä pitkiä. Valokuitu on myös immuuni sähkömagneettisille häiriöille, koska tieto siirretään valon avulla. Valokuitu ei myöskään yhdistä laitteita galvaanisesti eikä siinä ole ns. ”ylikuulumisen vaaraa”. [1, s.72]

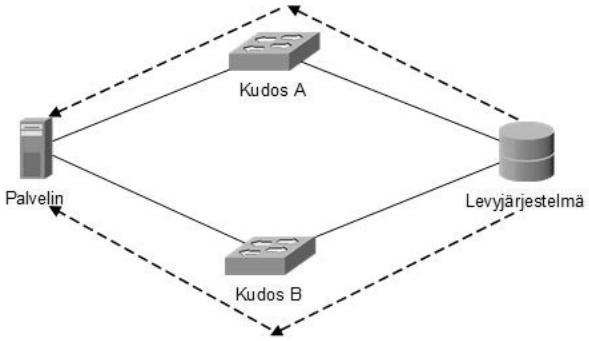
Tällä hetkellä markkinoilla olevan Fibre Channelin siirtonopeus on 8 Gbit/s, mikä tekee siitä nopean verrattuna jopa palvelimen sisäisiin I/O-väyliin tai lähiverkkoon kytkettyihin laitteisiin. Fibre Channelissa lähetys ja vastaanottaminen ovat mahdollisia samanaikaisesti (full duplex). Valokuitukaapeli sisältää aina kaksi rinnakkaista kuitua, joista toista käytetään lähettämiseen ja toista vastaanottamiseen. [1, s. 71; 3, s. 11]

2.2.2 Käyttöaste ja saatavuus

Tallennusverkko mahdollistaa fyysisen levytilan jakamisen usean palvelimen kesken, mikä vähentää esimerkiksi redundanttisen tiedon tarvetta useassa paikassa. Fyysinen levytila voidaan käyttää myös tehokkaammin hyödyksi, koska palvelimien levytilaa voidaan hallita keskitetysti.

Käyttämällä tallennusverkkoa paikallisten kiintolevyjen sijaan voidaan merkittävästi parantaa tiedon saatavuutta. Tämä saavutetaan kahdentamalla verkon eri osia, jotka voivat vaihdella palvelimen kuitukorttien (HBA) lisäämisestä koko datakeskuksen kahdentamiseen (remote backup-site). Mitä kriittisemmästä järjestelmästä on kyse, sitä

parempi tiedon saatavuus pitää olla. [3, s. 11] Yleinen ongelma tallennusverkoissa on I/O-väylän katkeaminen palvelimen ja levyjärjestelmän välillä. I/O-väylän herkin osa on valokuitukaapeli, joka vaurioituu erittäin herkästi. Pienikin fyysinen vaurio kuidussa estää valoa heijastumasta oikein kuidun sisällä, mikä johtaa koko väylän katkeamiseen tai ainakin merkittävään virheiden kasvuun. Tämä voidaan helposti välttää käyttämällä kahta tai useampaa väylää laitteiden välillä. Tallennusverkoissa kahden I/O-väylän käyttäminen vaatii tallennusverkon kahdentamista. Helpoiten tämä saavutetaan käyttämällä kahta kuituporttia jokaisessa palvelimessa. Kuituportit yhdistetään kahteen eri kytkimeen, joista molemmista menee kuitu tallennuslaitteeseen (kuva 2.2). [1, s. 207]

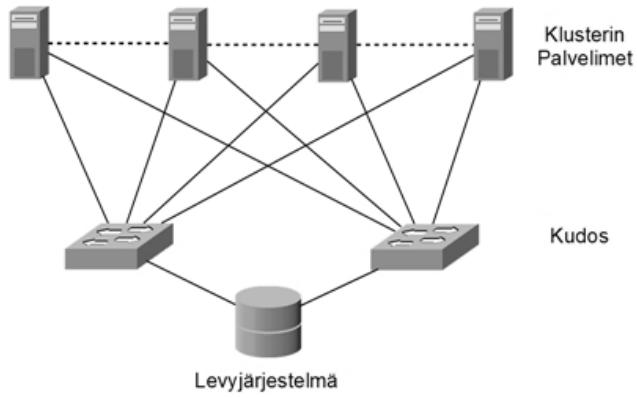


Kuva 2.2. Kaksi I/O-väylää palvelimen ja tallennuslaitteen välillä

Useamman I/O-väylän käyttäminen vaatii erillisen monikanavasovelluksen (multipath) käyttämistä, jotta palvelimelle näytetyt fyysiset levyt tunnistetaan samoiksi molempia kanavia pitkin. Monikanavointi ei löydy tällä hetkellä vielä Fibre Channel -standardista, mutta tulevaisuudessa se on tarkoitus lisätä FC-3 kerrokselle. Tämän hetkisissä järjestelmissä joudutaan käyttämään erillistä sovellusta, joka voidaan sijoittaa useaan eri kohtaan joko käyttöjärjestelmä- tai laitetasolla. [1, s. 208] Kahdennettu I/O-väylä voi toimia eri tavoilla.

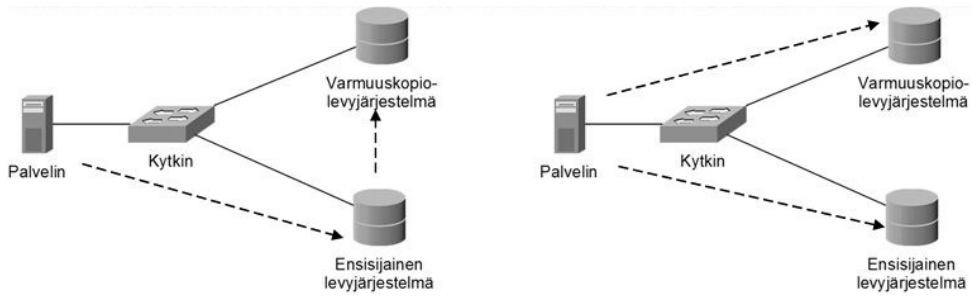
- *Aktiivinen/passiivinen*: molemmat väylät on yhdistetty levyjärjestelmään, mutta ainoastaan ensisijaista käytetään tiedonsiirtoon. Toissijaista väylää ei käytetä, ellei ensisijainen väylä hajoa.
- *Aktiivinen/aktiivinen*: Molempia väyliä käytetään tiedonsiirtoon, mutta levyjärjestelmästä palvelimelle osoitetut levyt on jaettu kahteen ryhmään, joista toista ryhmää käytetään ensimmäisen väylän kautta ja toista ryhmää toisen väylän kautta. Jos toinen väylistä rikkoutuu, kulkee kaikki tieto ainoastaan toisen väylän kautta.
- *Aktiivinen/aktiivinen (kuormantasaus)*: molempia kanavia käytetään kaikkien levyjen tiedonsiirtoon. Erona edelliseen on se, että kuorma pystytään tasamaan dynaamisesti molemmille väylälle. Jos toinen kanavista hajoaa, kaikki tieto siirretään ainoastaan yhtä kanavaa pitkin. [1, s. 20]

Tallennusverkon avulla on mahdollista tehdä palvelin-klustereita. Klusterit ovat menetelmä, jolla kaksi tai useampi palvelin saadaan näyttämään yhdeltä suurelta palvelimelta. Klustereiden avulla voidaan ehkäistä tilanteita, joissa kokonainen palvelin hajoaa. Klusterin palvelimet monitoroivat jatkuvasti toistensa tilannetta niin sanotun heartbeat-protokollan avulla. Toteutuksesta riippuen palvelimet lähettävät keepalive-viestejä toisilleen lähiverkon tai tallennusverkon kautta, jonka avulla klusterin palvelimet tietävät toistensa tilan. Samaan klusteriin kuuluvissa palvelimissa ajetaan samoja sovelluksia ja palvelimet käyttävät samoja fyysisiä levyjä, jotka sijaitsevat tallennusverkon palvelimilla (kuva 2.3). Ensisijaisen palvelimen hajoaminen käynnistää suorituksen toisella palvelimella. Käyttäjälle muutos on täysin läpinäkyvä. Klusterointia voidaan käyttää myös kuormanjakamiseen, jolloin kaikki klusterin palvelimet ovat aktiivisia samanaikaisesti. [2, sivu 199-200; 3, s.11]



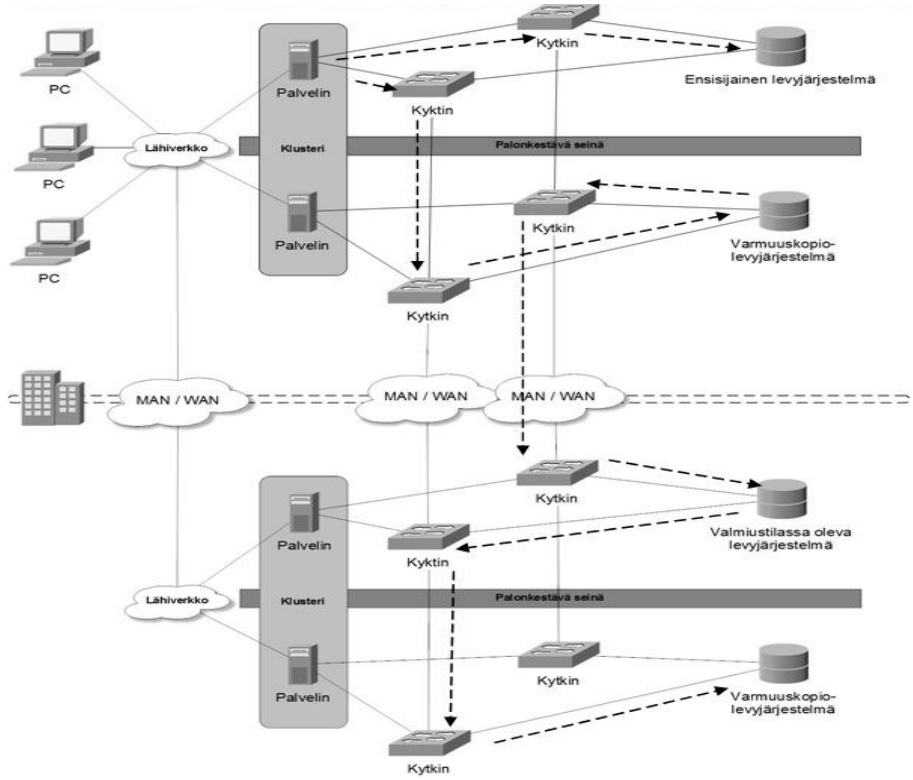
Kuva 2.3. Tallennusverkon käyttö mahdollistaa palvelimien klusteroinnin

RAID-tekniikan avulla pystytään välttämään tiedon menetys, jos yksittäinen kiintolevy hajoaa. Tästä ei kuitenkaan ole apua, jos koko levyjärjestelmä rikkoutuu. Ainoa ratkaisu levyjärjestelmän rikkoutumista vastaan on kiintolevyjen peilaaminen toiseen levyjärjestelmään. Peilaus voidaan suorittaa kahdella toisistaan poikkeavalla tavalla. Ensimmäisessä tavassa palvelin ei osallistu tiedon peilaukseen vaan peilaus tapahtuu tallennusverkossa kahden levyjärjestelmän välillä. Voidaan ajatella, että levyjärjestelmien tiettyjen fyysisten levyjen välillä on määritelty RAID 1-konfiguraatio, jolloin tieto peilaantuu toiseen levyjärjestelmään (kuva 2.4 vasemmalla). Tästä tavasta käytetään nimitystä ”remote mirroring”. [1, s. 212]



Kuva 2.4. Kaksi eri tapa peilata tieto kahdelle levyjärjestelmälle

Edellä mainitun tavan hyvä puoli on se, että se ei kuormita palvelinta. Tähän tapaan liittyy kuitenkin ongelma. Jos data korruptoituu tallennusverkossa palvelimen ja levyjärjestelmän välillä, virheet peilautuvat myös toiseen levyjärjestelmään. Näin molemmat levyjärjestelmät sisältävät virheellistä tietoa. Ongelma voidaan välttää siirtämällä peilaus palvelimelle loogisten levyjen hallintajärjestelmälle (volume manager). Tässä tavassa tieto siirretään kahdelle eri levyjärjestelmälle käyttäen kahta eri I/O-väylää (kuva 2.4 oikealla). Suurella todennäköisyydellä tieto säilyy eheänä ainakin toisessa levyjärjestelmässä. Ongelmaksi saattaa kuitenkin muodostua palvelimen ja tallennusverkon kuormittaminen, koska siirretävän datan määrä verkossa kaksinkertaistuu. [1, s. 213]



Kuva 2.5. Järjestelmä, jossa koko datakeskus on kahdennettu

Yhdistämällä edellä mainittuja kahdennustekniikoita saadaan aikaiseksi todella varmoja ratkaisuja järjestelmiin, joissa tiedon saatavuus on kaikkein tärkeintä. Edellä mainituista kahdennustekniikoista ei ole hyötyä, jos koko datakeskus tuhoutuu ennalta arvaamattomasti jonkin katastrofin vaikutuksesta. Keskitettyä tallennusratkaisua käyttämällä katastrofeihin on helpompi valmistautua kuin jos käytettäisiin keskittämätöntä tallennusta. DWDM-tekniikalla ja IP-protokollaan perustuvilla tallennusverkoilla levyjärjestelmät ja nauhakirjastot voidaan sijoittaa todella kauas datakeskuksista. Näillä tekniikoilla voidaan tehdä varmuuskopioita useiden satojen kilometrien päähän. Se miten nopeasti tallennettu tieto pitää olla saatavissa katastrofin jälkeen, riippuu datan tärkeydestä. Mikäli organisaatio pystyy jatkamaan toimintaansa päiviä ilman tallennusverkkoon tallennettua dataa, sille riittää datan ”manuaalinen”

palauttaminen esimerkiksi datanauhoilta. Toisaalta, jos tiedon saatavuus saa kärsiä ainoastaan muutamia sekunteja, täytyy koko datakeskus kahdentaa (kuva 2.5). Tällaisten järjestelmän rakentaminen ei missään tapauksessa ole halpaa, ja niitä käyttävätkin ainoastaan hyvin suuret yritykset ja valtiolliset elimet. Kahdennettua datakeskusta käytetään sovelluksissa, joissa saatavuuden katkeaminen ei ole vaihtoehto. Tällaisia järjestelmiä ovat esimerkiksi sotilaalliset, valtakunnan infrastruktuureja tai lentoliikennettä valvovat sovellukset. [1, s. 215; 2, s. 276-277; 2, s. 141]

2.2.3 Ylläpitokustannukset

Suurissa datakeskuksissa tallennusverkon käyttöön siirtymisellä saattaa olla suuret aloituskustannukset, mutta ne maksavat itsensä myös takaisin nopeasti. Tallennusverkon avulla voidaan datakeskuksen ylläpitokustannuksia vähentää merkittävästi, koska tiedon hallinta ja säilytys saadaan keskitettyä samaan kokonaisuuteen. Keskitetyt ratkaisut vähentävät manuaalisen työn määrää ja näin vähentävät ylläpidon työtunteja. Tämän lisäksi palvelimien sisäisiä kiintolevyjä ei tarvita enää niin paljon, koska keskitetyllä tallennustilalla voidaan maksimoida palvelimien tarvitseman levytilan käyttöaste.

Vaikka kiintolevyjen tallennustiheys kasvaa jatkuvasti, palvelimen sisäisillä kiintolevyillä tallennustilaa saadaan varsin rajallisesti. Kiintolevyt vaativat myös paljon tilaa, mikä kasvattaa palvelimen fyysistä kokoa. Suuret palvelimet vievät paljon lattiapinta-alaa datakeskuksessa [3, s. 10]. Mitä enemmän datakeskus vaatii pinta-alaa, sitä enemmän siitä aiheutuu myös kustannuksia. Yksi merkittävä kustannus datakeskuksissa aiheutuu tilojen jäähdytyksestä. Kiintolevyt ovat mekaanisia laitteita, jotka tuottavat paljon lämpöä. Keskitetyissä levyjärjestelmissä jäähdytys pystytään toteuttamaan tehokkaammin kuin yksittäisesti jokaisessa palvelimessa.

2.2.4 Skaalautuvuus

Käyttämällä palvelimen sisäisiä kiintolevyjä tai suoraan palvelimeen kiinnitettyjä tallennuslaitteita, tallennustilan kasvattaminen ilman palvelimen sammuttamista on vaikeaa. Käyttämällä tallennusverkkoa, palvelimille voidaan lisätä uutta tallennustilaa ilman, että palvelinta täytyy sammuttaa tai käynnistää uudelleen. Uudet levyt saadaan näkyviin palvelimelle skannaamalla SCSI-laitteet uudelleen.

Koko tallennusverkon kapasiteettiä pystytään myös kasvattamaan ajon aikaisesti, ilman että muita siihen liitettyjä laitteita tarvitsee ajaa alas. Datakeskuksen kasvaessa ja vaatimusten muuttumisen myötä tallennusverkkoon voidaan ongelmitta lisätä uusia palvelimia, kytkimiä ja tallennuslaitteita. [3, s. 11] Yritysten keskimääräinen tallennustilan määrä kaksinkertaistuu joka vuosi [1, s. 219].

Aiemmin mainitulla klusteritekniikalla yksittäiselle palvelimelle aiheutuvaa kuormaa voidaan vähentää lisäämällä klusteriin lisää palvelimia.

2.3 Tallennusverkon rajoitteet

Vaikka eri tallennusverkkotekniikoita on kehitetty jo pitkään, niihin liittyy edelleen monia rajoitteita, jotka vaikeuttavat tallennusverkkojen käyttöönottoa.

2.3.1 Standardointi

Yksi standardoinnin tavoitteista on helpottaa eri laitevalmistajien välisten laitteiden yhteensopivuutta. Ilman tekniikoiden standardointia laitteiden loppukäyttäjät joutuvat luottamaan yhteen laitetoimittajaan, jotta he voisivat olla varmoja IT-järjestelmän yhteensopivuudesta. Standardoinnista hyötyvät eniten loppukäyttäjät, koska silloin heidän ei tarvitse välttämättä luottaa yhden valmistajan tuotteisiin.

Tietokonejärjestelmien alemmat kerrokset, kuten verkon fyysinen kerros, kuljetus- ja hallintaprotokollat, on nykyään standardoitu, mutta prosessorit, käyttöjärjestelmät sekä ylemmän kerroksen sovellukset ovat edelleen hyvin valmistajakohtaisesti toteutettuja. Tallennusverkoissa klusterointi, varmuuskopiointi ja palautus, tallennustilan hallinta ja virtualisointi perustuvat standardoituun infrastruktuuriin, mutta sovellukset itsessään ovat hyvin valmistajakohtaisesti toteutettuja. Esimerkiksi varmuuskopiointi voidaan suorittaa tietyllä ohjelmalla, mutta sitä ei voida palauttaa käyttämällä toista ohjelmaa.

Tallennusverkkojen standardointi on muodostunut erittäin hankalaksi, koska se on hajonnut usealle standardointiorganisaatiolle. Esimerkiksi SCSI-protokolla on ANSI T10 komitean alainen standardi, FCP:n MIB, iSCSI, iFCP ja FCIP ovat IETF:n alaisia, Gigabit Ethernet on IEEE:n standardi, CIM on DMTF:n määrittelemä standardi, lisäksi tallennusverkkojen turvallisuudesta on standardeja ANSI T10:llä sekä IETF:llä. InfiniBandilla on myös oma standardointiorganisaatio IBTA. Kaiken tämän lisäksi on vielä SNIA, jonka tehtäviä ovat tallennusverkkojen virtualisoinnin ja CIM/WBEM/SMI-S hallintaprotokollien kehittäminen. Standardointiorganisaatioista harva toimii itsenäisesti. Valmistajat haluavat ottaa osaa myös standardointiin valvoakseen omia etujaan. Kaikki edellä mainittu johtaa liian monimutkaiseen ja hitaaseen standardointiprosessiin mistä lopulta kärsivät tallennusverkkojen käyttäjät. [2, s. 283-284]

2.3.2 Yhteensopivuus

Standardoinnista riippumatta eri valmistajien laitteiden välinen yhteensopimattomuus on yksi tallennusverkkojen suurimmista ongelmista. Vaikka valmistajien tuotteet perustuvat hyväksyttyihin standardeihin, ne eivät poista erilaisia toteutustapoja. Satojen eri valmistajien välisen yhteensopivuuden testaaminen on lähes mahdotonta, koska erilaisten konfiguraatioiden määrä nousee erittäin suureksi, kun otetaan huomioon kaikki laitteet ja niiden ohjelmistot sekä eri ohjelmistoversiot.

Valmistajat yleensä julkaisevat taulukoita (support matrix), joissa on lueteltuna testatut ja tuetut konfiguraatiot. Tällaisten listojen ylläpitäminen ajan tasalla on todella vaikeaa, koska valmistajat julkaiset uusia laitteita ja ohjelmistoversioita jatkuvasti. Loppukäyttäjien on pitäydyttävä valmistajan tukemissa konfiguraatioissa, mikäli he haluavat, että he saavat myös tukea valmistajalta ongelmatilanteissa. Tämä rajoittaa loppukäyttäjien valinnanvapautta merkittävästi ja tekee loppukäyttäjistä riippuvaisen yhdestä valmistajasta. [1, s. 168; 2, s. 286-287]

2.3.3 Hallinta

Tallennusverkot ovat erittäin monimutkaisia järjestelmiä, mikä tekee niiden hallinnasta erittäin vaikeaa. Tallennusverkot koostuvat kuitukorteista, porteista, kytkimistä, keskittimistä, silloista, nauhakirjastoista ja levyjärjestelmistä, joilla kaikilla on oma hallintasovellus laitteen konfigurointiin ja monitorointiin [2, s. 288]. Eri valmistajien tuotteiden lisääminen nostaa edelleen erilaisten hallintasovellusten määrää. Tällaisen järjestelmän hallinta on aikaa vievää, koska jokainen laite pitää konfiguroida erikseen omalla sovelluksella.

Keskitetyn hallintasovelluksen tekeminen on haastavaa, mikäli laitespesifiset hallintasovellukset eivät käytä samaa tapaa konfiguraatioiden tekemiseen ja tilastotietojen esittämiseen monitorointia varten. Nykyään SNIA:n kehittämä CIM, WBEM ja SMI-S ovat nousseet vallitseviksi protokolliksi tallennusverkkojen hallintaan, mikä on mahdollistanut useita valmistajia tukevien keskitettyjen hallintasovellusten kehittämisen tallennusverkkoihin. Tallennusverkkojen hallinnasta puhutaan enemmän luvussa 2.7.

2.3.4 Eri toteutusten yhdistäminen

Tallennusverkkoja voidaan toteuttaa nykyään monella eri tavalla. Fibre Channeliin perustuvat ratkaisut ovat nousseet suosituimmaksi tavaksi, mutta IP-protokollaan ja Ethernet-tekniikkaan (FCoE) perustuvat ratkaisut ovat herättäneet mielenkiintoa viime aikoina. Näiden lisäksi tallennusverkko voidaan toteuttaa InfiniBand:n avulla. Erilaisten tallennusverkkojen yhdistäminen yhdeksi kokonaisuudeksi vaatii jälleen uusien teknologioiden valjastamista. Eri tavoista toteuttaa tallennusverkko kerrotaan lisää luvussa 2.8. [2, s. 289]

Eri tallennusverkkojen toteutustapoja voidaan yhdistää toisiinsa esimerkiksi useaa eri protokollaa tukevilla kytkimillä tai tallennuslaiteilla sekä SAN-silloilla. Tallennusverkon laitteista kerrotaan lisää luvussa 2.4

2.3.5 Hinta

Tallennusverkon hyödyt alkavat tulla esille vasta kun palvelimia on vähintään kymmeniä. Mitä enemmän palvelimia datakeskuksessa on sitä enemmän tarve

tallennusverkon käyttöön kasvaa. Tallennusverkon laitteet ovat valitettavasti edelleen todella kalliita, mikä tekee niiden käytön monille organisaatiolle mahdottomaksi. IP- ja Ethernet-tekniikkaan perustuvien tallennusverkkojen uskotaan tuovan hintoja alaspäin tulevaisuudessa.

2.4 Tallennusverkon arkkitehtuuri ja laitteet

Tallennusverkot koostuvat useista eri laitteista, jotka voidaan karkeasti jakaa kolmeen eri kerrokseen. Nämä kerrokset ovat palvelin- (host), kudos- (fabric) ja tallennuskerros (storage).

Palvelin-kerros koostuu itse palvelimesta ja sen lisälaitteista, jotka mahdollistavat pääsyn tallennusverkkoon. Fibre Channel -tekniikalla toteutetussa tallennusverkossa palvelin tarvitsee yhden tai useamman kuitukortin (HBA) ja sen portteihin lisättävän GBIC-moduulin, joka muuntaa sähköisen signaaliin optiseksi signaalisiksi ja toisin päin. Kudoskerrokselle kuuluvat laitteet, jotka yhdistävät päätelaitteiteet (palvelimet ja tallennuslaitteet) ja verkon osat toisiinsa ja ohjaavat liikenteen oikeaan suuntaan. Tällaisia laitteita ovat keskittimet, kytkimet, sillat sekä reitittimet. Tallennuskerroksella sijaitsevat levyjärjestelmät sekä nauhakirjastot, jotka toimivat keskitettyinä massamuistilaitteina palvelimille.

2.4.1 Palvelimet

Palvelin tarvitsee lisälaitteen, jotta se voi hyödyntää erillistä tallennusverkossa sijaitsevaa massamuistilaitetta. Kuitukortti on laite, joka yhdistää palvelimen kudokseen. Kuitukortti tarvitsee laiteohjaimen (device driver), jonka avulla käyttöjärjestelmä pystyy käyttämään tätä lisälaitetta. Lisälaitteen ja sen ohjaimen täytyy pystyä piilottamaan tallennusverkon takana oleva fyysinen levytila käyttöjärjestelmältä, siten että tallennustila näyttää käyttöjärjestelmälle paikalliselta kiintolevyltä.

GBIC

Tallennusverkon laitteiden portit muuntavat signaalin optisesta sähköiseen muotoon ja toisin päin. Tästä toiminnosta vastaa porttiin liitettävä laite nimeltä GBIC (kuva 2.6). Tästä laitteesta on käytetty aiemmin myös lyhennettä GLM. GBIC on nykyään erillinen moduuli, joka voidaan vaihtaa tarvittaessa. GBICit voidaan jaotella kahteen luokkaan, riippuen niiden käyttämästä valon aallonpituudesta. Lyhyttä aallonpituutta käyttävien GBICien valon aallonpituus on 780-850 nm, jota voidaan käyttää alle 500 metriä pitkissä kuitukaapeleissa. Pitkää aallonpituutta käyttävien GBICien aallonpituus on 1300 nm, jonka avulla kuitukaapelin pituus voi olla jopa satoja kilometrejä. Jokaisessa GBIC-moduulissa on kaksi liitintä, joista toista käytetään datan vastaanottamiseen ja toista datan lähettämiseen. Tämä mahdollistaa datan samanaikaisen vastaanottamisen ja lähettämisen (full duplex). [3, s. 27-28]

Kuva 2.6. GBIC

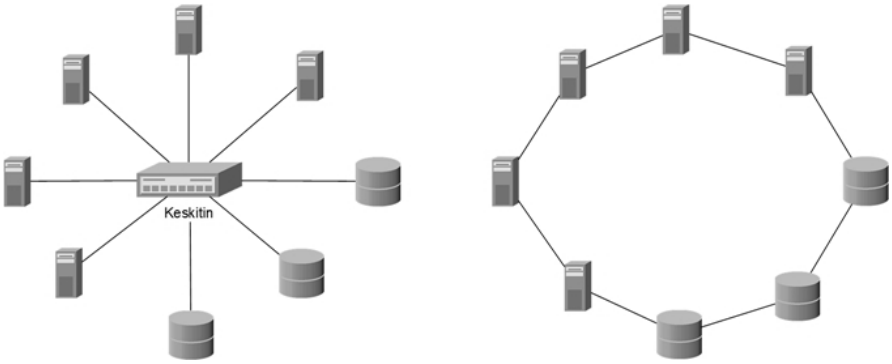
Toisen sukupolven GBIC-teknologia mahdollistaa hallinta- ja monitorointitiedon keräämisen tallennusverkon fyysiseltä kerrokselta. Laite, johon GBIC on kytketty (keskitin, kytkin, palvelin) voi kerätä tilastotietoja tai tietoa GBIC:n tukemista ominaisuuksista ja välittää tätä tietoa eteenpäin esimerkiksi tallennusverkkoa monitoroivalla sovellukselle. GBIC:stä saatavaa tietoa ovat mm. valmistaja, sarjanumero, tuetut nopeudet, tehonkulutus ja muut fyysisen kerroksen diagnostiikkatiedot. [2, s. 106]

2.4.2 Kudoskerros

Kytkemällä palvelimet suoraan tallennuslaitteisiin kulutetaan todella paljon portteja. Palvelimia voi liittää tallennuslaitteeseen vain niin monta kuin siinä on portteja. Tämä ratkaisu ei eroa paljoa suoraan palvelimeen kytketystä tallennustilasta, eikä se muistuta verkkoa millään tavalla. Tallennusverkon luomiseen tarvitaan laite, joka yhdistää laitteet toisiinsa ja ohjaa liikennettä niiden välillä. Tallennusverkossa näistä toiminnoista vastaavat keskittimet, kytkimet, sillat sekä reitittimet.

Keskittimet

Keskitin (Hub) on yksinkertainen passiivinen elektroninen laite, johon tallennusverkon laitteet kytketään kuitukaapelilla. Käyttämällä keskitintä verkon fyysinen rakenne on tähtimäinen, mutta loogisesti verkosta muodostuu rengas (kuva 2.7). Keskitin ohjaa tulevan liikenteen aina seuraavaan porttiin, jossa on laite kytkettynä. Jokainen keskittimeen kytketty laite lähettää vuorollaan datan eteenpäin, mikäli se ei ole vastaanottaja. Data kiertää läpi verkon kunnes se saavuttaa vastaanottajan. Rengasrakenteesta johtuen, ainoastaan yksi laite voi käyttää verkkoa samanaikaisesti (half duplex). [3, s. 15] Tallennusverkon keskitin jakaa väylän kaikkien siihen liitettyjen laitteiden kesken eli se on analoginen lähiverkoissa käytettävien moniporttitoistimien kanssa. Erona näiden välillä on kuitenkin se, että lähiverkoissa looginen rakenne ei ole rengas vaan jaettu väylä. Tallennusverkon ja lähiverkon keskittimet jakavat myös tietoturvaan liittyvän ongelman. Lähiverkon keskitin lähettää kaiken liikenteen kaikille siihen kytketyille laitteille. Tallennusverkoissa liikenne on nähtävissä kaikilla laitteilla, jotka osallistuvat tiedon välitykseen lähettäjän ja vastaanottajan välillä.



Kuva 2.7. Keskittimen avulla muodostetun tallennusverkon fyysinen (vas.) ja looginen (oik.) topologia

Keskittimen huono puoli on se, että käytössä oleva kaistanleveys jakautuu kaikkien siihen liitettyjen laitteiden kesken. Mitä enemmän renkaaseen on liitetty laitteita, sitä vähemmän yksittäinen laite voi käyttää koko kaistanleveydestä. Tästä syystä keskittimiä ei voi käyttää tallennusverkoissa, joissa liikennemäärät ovat suuria. Rengas-rakeenteessa käytettävän Fibre Channel -protokollan versio FC-AL määrittelee, että yhteen renkaaseen voidaan liittää ainoastaan 126 laitetta, mikä rajoittaa merkittävästi tallennusverkon kokoa [7, s. 12]. Tosin näin monen laitteen kilpaileminen samasta kaistasta tukkisi verkon melko nopeasti. [3, s. 30-33]

Keskittimet voidaan jakaa kolmeen eri luokkaan. Halvimmat keskittimet ovat edellä mainittuja passiivisia verkkolaitteita, jotka ainoastaan muodostavat rengastopologian laitteiden välille. Ne eivät tarjoa hallinta- tai monitorointipalveluja ylläpitäjille. Hallittavat keskittimet tarjoavat ylläpitäjille mahdollisuuden kerätä diagnostiikkatietoa keskittimestä esimerkiksi web-sovelluksen, telnetin tai SNMP:n avulla. Näiden avulla ylläpitäjät saavat tietoa esimerkiksi laitteiston toiminnasta ja kuormitusasteesta. Kytkevät keskittimet asettuvat nimensä mukaisesti keskittimien ja kytkimien välimaastoon. Kytkevät keskittimet eivät muodosta kudosta eikä näin ollen tarjoa kudoksen palveluita, mutta mahdollistavat full duplex-liikenteen. Keskittimen portit voidaan jakaa useisiin virtuaalisiin renkaisiin. Kriittisimmät portit voidaan asettaa yksin omaan renkaaseensa. [1, s. 99]

Tallennusverkon keskittimet ovat nopeasti poistuneet markkinoilta kytkimien yleistyessä ja niiden hinnan pudotessa. Keskittimet ovat kuitenkin halpa ratkaisu tallennusverkkoihin, joissa liikennemäärät pysyvät pieninä ja liikenteen näkyminen muille renkaan laitteille ei ole ongelma. [2, s. 118]

Kytkimet

Kytkin ei ulkoisesti eroa merkittävästi keskittimestä (kuva 2.8). Fyysinen rakenne on tähtimäinen kuten keskitintäkin käyttämällä. Kytkin on kuitenkin huomattavasti älykkäämpi laite kuin keskitin ja sen avulla myös looginen rakenne pysyy tähtimäisenä. Kytkin mahdollistaa full duplex-liikenteen kaikkien siihen kytkettyjen laitteiden välillä samanaikaisesti. [3, s. 15]

Kuva 2.8. Kaksi Tallennusverkon kytkintä

Tallennusverkon kytkin on kudoksen komentokeskus, jonka älykkyys perustuu sen tuottamiin kudoksen palveluluihin, joita ovat mm. pakettien reititys, nimipalvelu, zoning-palvelu sekä aliaksien muodostaminen. [1, s. 97]

Kytkimet tukevat niin sanottua ”cut-through” kytkentää sekä pakettien puskurointia [1, s. 97]. Cut-through-kytkentä on menetelmä, jossa kytkin reitittää paketin jo ennen kuin se on kokonaan saapunut perille. Kytkin ohjaa paketin oikeaan suuntaan välittömästi, kun se on saanut paketin otsikosta kohdeosoitteen selville. Menetelmä vähentää kytkennän aiheuttamaa viivettä, mutta vaatii paljon tehoa kytkimeltä. Pakettien puskuroinnissa saapuvat paketit asetetaan kytkimen välimuistiin kunnes ne on kokonaan vastaanotettu, jonka jälkeen paketti reititetään eteenpäin. Menetelmä aiheuttaa viivettä kytkentään, mutta on turvallisempi kuin cut-through-kytkentä, koska paketin eheys pystytään tarkistamaan ennen sen kytkentää eteenpäin.

Nimipalvelu on yksi tärkeimmistä kytkimen tuottamista palveluluista. Sen avulla kytkin tietää, mihin porttiin tietty laite on kytketty tallennusverkossa. Kun laite kytketään kytkimeen, se kirjautuu sisään kudokseen (fabric login). Laitteen osoite (WWN) rekisteröityy kytkimen nimipalvelimeen, jotta se jatkossa tietää mihin porttiin laite on kytketty. [3, s. 46-47]

Zoning on tunnetuin tallennusverkkojen tietoturvaa lisäävä palvelu, jonka avulla voidaan rajata laitteiden näkymistä toisilleen. Tekniikka on samankaltainen kuin lähiverkkojen VLAN-tekniikkaan, jolla voidaan jakaa lähiverkko useaan virtuaalisen aliverkkoon. Tallennusverkoissa tämä rajaus saadaan aikaiseksi lisäämällä halutut laitteet samaan vyöhykkeeseen (zone). Ainoastaan samaan vyöhykkeeseen kuuluvat laitteet voivat keskustella toistensa kanssa tallennusverkon yli. Laitteet voidaan lisätä vyöhykkeeseen pehmeällä tai kovalla tavalla. Pehmeässä tavassa kytkimen zone-palvelin (FCZS) jakaa siihen kytketyille laitteille listan osoitteista, jonka kanssa laite voi keskustella. Tämän tavan pehmeys perustuu siihen, että mikään ei kuitenkaan estä laitetta keskustelemasta muidenkin laitteiden kanssa. Kova vyöhykkeisiin jako tehdään lisäämällä kytkimen portti tai laitteen WWN-osoite tiettyyn vyöhykkeeseen. [4, s. 391-

392] Porttiin perustuvassa jaossa vyöhykejako pysyy samana vaikka laitetta portissa vaihdettaisiin. Jos laitteen paikkaa tallennusverkossa vaihdetaan, täytyy laitteen uusi portti vaihtaa vyöhykkeeseen. WWN-osoitteen lisääminen vyöhykkeeseen mahdollistaa laitteen paikan vaihtamisen, koska jako vyöhykkeisiin perustuu laitteen osoitteeseen. Tässä tavassa vyöhykekonfiguraatiota pitää muuttaa, jos laitteen osoite vaihtuu esimerkiksi rikkinäisen kuitukortin vaihdon takia.

Ylläpitäjien työn helpottamiseksi porteille ja WWN-osoitteille voidaan määritellä helpommin muistettavia nimiä, aliaksia. Aliaksia voidaan käyttää tallennusverkon hallinnassa, kuten esimerkiksi vyöhykejakojen tekemisessä. [1. s. 90].

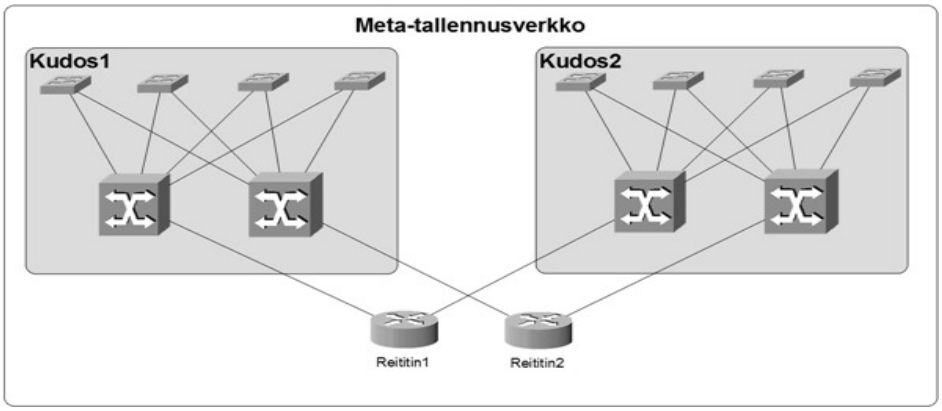
Tallennusverkon kytkimet voidaan karkeasti jakaa kahteen eri luokkaan. Tavalliset kytkimet sisältävät joitakin kymmeniä kuituportteja ja ne tarjoavat kaikki kudoksen tarvitsemat palvelut. Tällaiset kytkimet voivat sisältää joitakin kahdennettuja komponentteja, kuten teholähde tai tuuletin, mutta eivät varsinaisesti tarjoa korkeaa saatavuutta (HA). Tällaisia kytkimiä ei ole mielekästä käyttää suurissa tallennusverkoissa, koska liian moni kytkimen porteista joudutaan käyttämään kytkimien välisiin linkkeihin (ISL) [2, s. 126]. Pienemmät kytkimet sopivatkin eräänlaisen pääsyverkon muodostamiseen, joista liikenne kerätään verkon ytimeen. Verkon ytimessä käytettävistä suurista kytkimistä käytetään nimitystä ”director”. Director-luokan kytkimet sisältävät usein satoja kuituportteja, joita pystytään myös lisäämään ajon aikaisesti. Kaikki director-kytkimen komponentit on kahdennettu, jolla vältetään yksittäisen vikaantuneen komponentin aiheuttama verkon hajoaminen. Vikaantuneet komponentit pystytään myös vaihtamaan ajon aikaisesti. [2, s. 126]

Reitittimet ja sillat

Useista kytkimistä koostuvassa kudoksessa kaikki edellä mainittujen palveluiden tiedot pitää pystyä levittämään kaikille muille kytkimille, jotta laitteet löytävät toisensa tallennusverkossa. Kudoksen kasvaessa yhä suuremmaksi tähän kytkimien väliseen synkronointiin kuluva aika kasvaa merkittävästi. Hierarkiattoman verkon ongelmat ovat tuttuja esimerkiksi Ethernet-tekniikasta, jossa ongelma on ratkaistu pilkkomalla verkko pienempiin hallittaviin kokonaisuuksiin. Tallennusverkot ovat kohdanneet saman ongelman ja markkinoille on tullut laitteita, reitittimiä, joilla ongelma voidaan ratkaista myös tallennusverkoissa. [6, s. 4]

Fibre Channel -reititin on laite, jonka avulla voidaan yhdistää useita kudoksia toisiinsa, ilman että ne sulautuvat yhdeksi kudokseksi (kuva 2.9). Kahden tai useamman kudoksen yhdistäminen ilman reititintä aiheuttaa sen, että kaikkien kudosten nimipalvelut, FSPF-tietokannat ja vyöhykekonfiguraatiot yhdistyvät yhdeksi, muodostaen näin ainoastaan yhden kudoksen. Kudosten varomaton yhdistäminen saattaa johtaa suuriin ongelmiin. Päällekkäisyydet vyöhykekonfiguraatioissa, eroavaisuudet kudosten parametreissa sekä tietoturvapoliittiset konfliktit voivat johtaa odottamattomiin tilanteisiin. [6, s. 21-22]

Fibre Channel -reitittimen tehtävä on melko samanlainen kuin IP-verkon reitittimenkin. Ero näiden kahden välillä on se, että Fibre Channel -reititin on enemmänkin kytkevä palomuuri kuin reititin. FC-reititin ei avaa vapaata liikennettä kahden kudoksen välillä kuten IP-reititin kahden Ethernet-aliverkon välillä, vaan sallii ainoastaan määrätyn liikenteen kudosten välillä. [6. s. 22]



Kuva 2.9. Tallennusverkon reititin muodostaa meta-tallennusverkon

Tallennusverkoista puhuttaessa reititin ei ole vakiintunut nimitys laitteelle, joka yhdistää kudoksia tai eri tallennusverkkoratkaisuja toisiinsa. Reitittimen (router) lisäksi tällaisesta laitteesta voidaan käyttää myös nimeä silta (bridge) tai yhdyskäytävä (gateway). Silta nimenä sopii tilanteisiin, joissa halutaan yhdistää kaksi erilaista tallennusratkaisua toisiinsa. Silta muuttaa käytettävän protokollan toiseksi, jotta esimerkiksi vanha SCSI-levyjärjestelmä ja Fibre Channel-protokollalla toteutetussa tallennusverkossa oleva palvelin voidaan yhdistää toisiinsa. IP-protokollaa käyttävillä tallennusverkon reitittimillä voidaan myös yhdistää FC-protokollaa käyttäviä kudoksia toisiinsa Internetin yli, jolloin yhdyskäytävä on sopiva nimi. [3, s. 38]

2.4.3 Tallennuskerros

Kiintolevyt ja datanauhat ovat tällä hetkellä suosituin tapa tallentaa tietoa. Tallennusverkkoa käytettäessä palvelimien sisäiset kiintolevyt tai DAS-laitteet korvataan keskitetyllä tallennuslaitteella. Tällaiset keskitetyt laitteet sisältävät sadoista gigatavuista aina useaan petatavuun asti tallennustilaa. Tämän lisäksi ne tarjoavat parempaa saatavuutta ja skaalautuvuutta, kehittyneempiä varmuuskopiotekniikoita ja parempaa tallennustilan käyttöastetta verrattuna keskittämättömään ratkaisuun. [1, s.15]

Levyjärjestelmät

Levyjärjestelmää voidaan pitää eräänlaisena kiintolevypalvelimena, johon toiset palvelimet yhdistyvät tallennusverkon kautta käyttämällä esimerkiksi SCSI- ja Fibre Channel-protokollaa. Levyjärjestelmän tarkoitus on korvata palvelimien sisäiset kiintolevyt ja vanhat SCSI-tallennuslaitteet keskitetyllä tallennusverkkoon kytkettävällä

massamuistilaitteella, joka tallennustilan lisäksi tarjoaa parempaa suorituskykyä, luotettavuutta, saatavuutta ja tehokkaita varmuuskopiointipalveluita.

Levyjärjestelmien voidaan karkeasti olettaa koostuvan I/O-porteista, ohjaimista, välimuisteista, sisäisistä I/O-väylistä sekä itse kiintolevyistä. I/O-portit liittävät levyjärjestelmän suoraan palvelimiin tai tallennusverkkoon. I/O-portit jatkavat levyjärjestelmän sisäistä I/O-väylää palvelimille asti, jotta palvelimet näkevät niille tarkoitetut fyysiset tai loogiset levyt. Suurin osa levyjärjestelmistä sisältää ohjaimen (controller), joka yhdistää I/O-portit ja kiintolevyt toisiinsa. Ohjain on levyjärjestelmän älykäs laite, jonka avulla kiintolevyjen saatavuutta ja luotettavuutta voidaan parantaa mm. RAID-tekniikan avulla. [1, s. 16] Levyjärjestelmät voidaan luokitella kolmeen eri luokkaan ohjaimen perusteella. Ensimmäiseen luokkaan kuuluvat levyjärjestelmät, joissa ei ole ohjainta. Tällaiset laitteet ovat vain erillisiä kiintolevykehikoita (JBOD). Palvelimet tunnistavat jokaisen levyn omana laitteenaan, jonka takia niillä täytyy jokaisella olla oma osoite. Toiseen luokkaan kuuluvat levyjärjestelmät, joilla on RAID-tekniikkaa tukeva ohjain. Kolmanteen luokkaan kuuluvat älykkäät ohjaimet, jotka tukevat tallennusverkon lisäpalveluita, kuten varmuuskopiointia. [1, s. 21]

Välimuistia käytetään tietokonejärjestelmissä nopeuttamaan hitaita operaatioita. Kiintolevyt ovat tunnetusti hitaita laitteita verrattuna muihin tietokoneen osiin. Levyjärjestelmissä välimuistia käytetään luku- ja kirjoitusoperaatioiden nopeuttamiseen. Välimuistia levyjärjestelmissä on sekä itse fyysisissä kiintolevyissä että ohjaimessa. Levyjärjestelmän I/O-väylä on paljon nopeampi kuin kiintolevyjen luku- ja kirjoitusnopeus. Tästä syystä kiintolevy tallentaa kirjoitettavan datan ensin omaan välimuistiinsa, josta se kirjoittaa datan kiintolevylle kun sillä on aikaa. Lukuoperaatioissa levy siirtää luettavan datan ensin välimuistiin, josta ohjain vasta käy sen lukemassa. [1, s. 40]

Suurempien levyjärjestelmien ohjaimet sisältävät yleensä paljon välimuistia. Palvelimien kirjoitusoperaatiot ovat yleensä purskeista, jolloin suuria määriä dataa kirjoitetaan kerralla. Ohjain tallentaa kirjoitettavan datan välimuistiin, josta se kirjoittaa sen levyille. Kirjoitusvälimuisti on hyvä olla varmennettu patterilla, jotta sähkökatkoksen yllättäessä ei menetetä dataa. Lukuvälimuistin toiminta on hieman monimutkaisempaa. Ohjaimen pitää osata ennakoida, mitä dataa palvelin haluaa seuraavaksi. Ohjain siirtää dataa valmiiksi välimuistiin, jotta se voi nopeammin välittää sen palvelimelle. [1, s. 41]

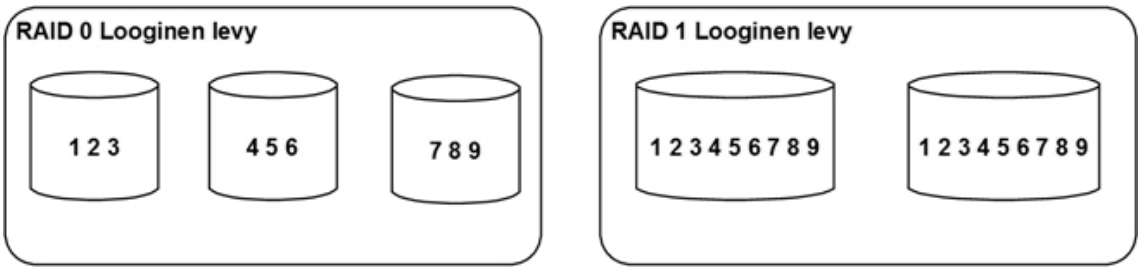
Levyjärjestelmissä käytetään useita virtualisointitekniikoita. Tärkein näistä tekniikoista on RAID-tekniikka, jonka avulla useasta kiintolevystä voidaan muodostaa yksittäisiä loogisia kokonaisuuksia. Palvelimelle RAID:n muodostama looginen levy näkyy ainoastaan yhtenä kiintolevynä. RAID-tekniikalla on kaksi tavoitetta. Ensimmäinen tavoite on suorituskyvyn kasvattaminen juovittamalla (striping) data usealle kiintolevylle, jolloin luku- ja kirjoitusoperaatiot saadaan jaettua usean kiintolevyn

kesken. Kiintolevyt ovat mekaanisia laitteita, joiden nopeus riippuu levyjen pyörimisnopeudesta, sekä luku/kirjoituspään liikkumisen aiheuttamasta viiveestä. Yksittäisen kiintolevyn käyttäminen tallennustilana muodostuu usein suorituskyvyn pullonkaulaksi. RAID-tekniikalla suorituskykyä voidaan parantaa jakamalla data palasiksi usealle kiintolevylle, jolloin jokainen kiintolevy osallistuu levyoperaatiohin. Toinen RAID-tekniikan tavoite on luotettavuuden lisääminen peilaamalla data usealle kiintolevylle. RAID-tekniikasta on useita eri tasoja, joilla saadaan aikaiseksi erilaisia suorituskyky- ja luotettavuusasteita. [1, s. 22]

RAID-tasolla nolla levyjärjestelmän ohjain juovittaa palvelimen lähettämän kirjoitettavan datan usealle fyysiselle kiintolevylle (kuva 2.10 vasemmalla). RAID-tason nolla avulla voidaan siis kasvattaa palvelimelle näkyvän kiintolevyn suorituskykyä. Se ei kuitenkaan lisää kiintolevyn luotettavuutta, koska yhden kiintolevyn hajoaminen RAID 0-konfiguraatiossa aiheuttaa kaikkien siihen kuuluvien kiintolevyjen sisältämän datan menettämisen. Voidaan jopa sanoa, että RAID-taso nolla vähentää kiintolevyn luotettavuutta. On todennäköisempää, että useammasta kiintolevystä hajoaa yksi kuin, että yksi ainoa kiintolevy hajoaa. Pelkkää RAID-tasoa nolla käytetään sovelluksissa, joissa suorituskyky on tärkeämpää kuin datan säilyminen.

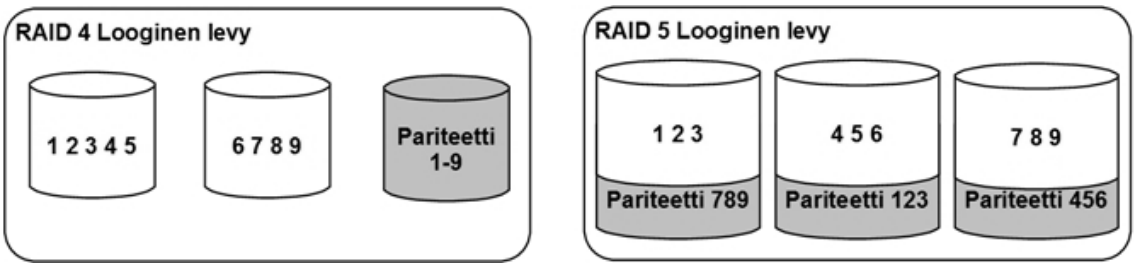
RAID-tasolla yksi levyjärjestelmän ohjain peilaa palvelimen lähettämän kirjoitettavan datan usealle fyysiselle kiintolevylle (kuva 2.10 oikealla). Sen avulla voidaan lisätä palvelimelle näkyvän kiintolevyn luotettavuutta, mutta ei suorituskykyä. Eli jos palvelin haluaa kirjoittaa kiintolevylle, levyjärjestelmän ohjain kirjoittaa saman datan kahdelle tai useammalle eri kiintolevylle. Pelkkää RAID-tason yksi tekniikkaa käytetään tilanteissa, joissa pääpaino ei ole suorituskyvyllä vaan luotettavuudella. RAID-taso yksi ei ole kuitenkaan taloudellinen tapa ratkaista luotettavuusongelma, koska siinä palvelimelle näkyvä kapasiteetti on puolet vähemmän mitä se todellisuudessa käyttää levyjärjestelmästä [9, s. 211].

RAID-tasot nolla ja yksi voidaan yhdistää, jotta saadaan molemmista tasoista hyvät puolet. Yhdistäminen voidaan toteuttaa kahdella tavalla, joista käytetään nimityksiä RAID 0+1 ja RAID 10. RAID 0+1:ssä data peilataan ensin ja tämän jälkeen juovitetaan. RAID 10:ssä data juovitetaan ensin ja tämän jälkeen peilataan. RAID-tason yksi ongelma säilyy myös tasojen yhdistelmässä. Tallennustilaa hukataan ”turhaan” peilaamiseen. [1, s. 26–30]



Kuva 2.10. Datan kirjoittaminen fyysisille levyille RAID-tasoilla nolla ja yksi

RAID-tasoilla neljä ja viisi käytetään erilaista tekniikkaa, jolla saavutetaan sekä parempaa luotettavuutta että suorituskykyä. RAID-tasolla neljä palvelimen lähettämä data juovitetaan fyysisten kiintolevyjen kesken. Yhdelle kiintolevyistä tallennetaan datan sijasta pariteettibitit kyseisistä juovista (kuva 2.11 vasemmalla). Jos yksi fyysisistä kiintolevyistä hajoaa, voidaan data generoida muiden levyjen ja pariteettilevyn juovien avulla. RAID-taso neljä parantaa fyysisten levyjen käyttöastetta, koska redundantin datan tallentamiseen ei tarvita kaksinkertaista määrää kiintolevyjä. Lukuoperaatiot virtuaaliselta levyltä nopeutuvat RAID 4:n avulla, mutta kirjoittamiseen vaaditaan enemmän aikaa, koska pariteetti pitää laskea ja tallentaa erikseen. Lisäksi pariteettibittilevy saattaa muodostua järjestelmän pullonkaulaksi, koska pariteettibitit tallennetaan yhdelle kiintolevylle. Suuria datamääriä kirjoitettaessa tämä yksittäinen kiintolevy saattaa ylikuormittua. [9, s. 211] RAID-tasolla viisi tämä ongelma on ratkaistu kierrättämällä pariteettilohkoa jokaisen kiintolevyn kesken, jolloin yksi kiintolevy ei muodostu pullonkaulaksi (kuva 2.11 oikealla).



Kuva 2.11. Datan kirjoittaminen fyysisille levyille RAID-tasoilla neljä ja viisi

RAID-tasot neljä ja viisi selviytyvät yhden fyysisen kiintolevyn hajoamisesta. Mikäli RAID-konfiguraatiosta hajoaa toinenkin kiintolevy ennen kuin viallinen on ehditty vaihtaa, johtaa se datan menettämiseen. Viallisen kiintolevyn vaihtaminen vaatii valtavan määrän luku ja kirjoitusoperaatiota, koska RAID-konfiguraatio pitää palauttaa laskemalla pariteetit uudelleen. Tämä saattaa helposti johtaa toisen kiintolevyn hajoamiseen. RAID-tasossa kuusi konfiguraation lisätään toinen pariteettibittilevy, jolloin virtuaalinen levy selviää kahden fyysisen levyn hajoamisesta. Tämä kuitenkin pahentaa kirjoitusoperaatioista aiheutuvaa viivettä, koska pariteettibitit pitää tallentaa kahdelle kiintolevylle.

RAID-tasot kaksi ja kolme ovat poistuneet käytöstä jo kauan aikaa sitten, eivätkä ne ole enää tehokkaita nykyaikaisiin järjestelmiin. RAID-tasolle neljä on käymässä samoin tavoin. RAID-taso viisi on luku- ja kirjoitusoperaatioiden nopeuden ja pariteettibittien tuoman luotettavuutensa ansiosta noussut suosituimmaksi tekniikaksi toteuttaa luotettavia levyjä.

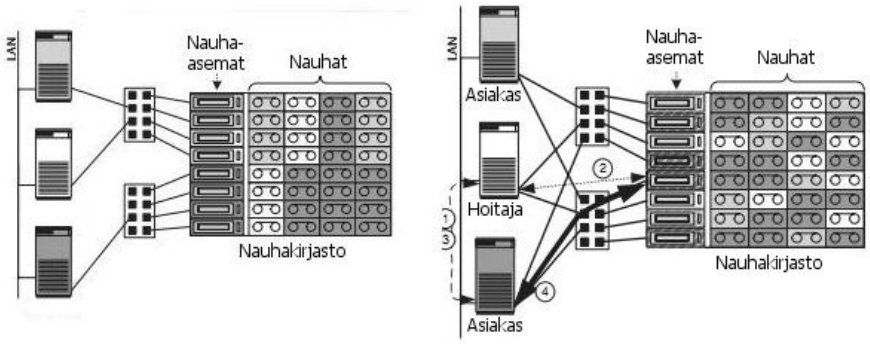
Levyjärjestelmä tarvitsee suodattimen, jolla estetään palvelimia näkemästä kaikkia virtuaalisia tai fyysisiä levyjä, joita levyjärjestelmä sisältää. Suodatus saadaan aikaiseksi liittämällä jokaiseen levyyn identifioiva numero, LUN. Ilman tällaista suodatusta, levyt täytyy konfiguroida palvelimien käyttöjärjestelmiin erittäin huolellisesti, jotta ne eivät

vahingossa kirjoita toistensa kiintolevyille ja tuhoa tallennettua dataa. LUN naamioinnin (masking) avulla levyjärjestelmä antaa palvelimen nähdä ainoastaan levyt, joita se tarvitsee. LUN-naamiointi voidaan tehdä portti- tai palvelin-perusteisesti. Portti-perusteinen naamiointi löytyy yleensä halvemmista levyjärjestelmistä. Siinä samaan levyjärjestelmän porttiin liitetyt palvelimet näkevät samat levyt. Palvelinperusteinen naamiointi on parempi, koska siinä naamiointia ei sidota mihinkään porttiin vaan esimerkiksi palvelimen WWN-osoitteeseen. [1, s. 51-54]

Nauhakirjastot

Nauhakirjastot (tape library) ovat tallennuslaitteita, joissa tieto tallennetaan magneettisten kiintolevyjen sijasta datanauhoille. Keskitettyjen levyjärjestelmien avulla halutaan pienentää palvelimen sisäisten kiintolevyjen määrää ja lisätä niiden luotettavuutta. Nauhakirjaston tarkoitus on vastaavasti korvata yksittäiset palvelimiin kiinnitettävät nauha-asemat keskitetyllä suuremmalla laitteella. Nauhakirjastot pystyvät käsittelemään useita nauhoja samanaikaisesti, joten niitä voidaan pitää useana erillisenä nauha-asemana. Nauhakirjastoja ei voi käyttää tiedon ensisijaisena tallennuspaikkana, koska nauhoilta lukeminen on vielä hitaampaa kuin kiintolevyiltä. Nauhatallennus on kustannustehokas tapa tiedon varmuuskopiointiin ja pitkäaikaiseen tallennukseen. [10, s. 16]

Nykyaikaiset nauhakirjastot voidaan liittää suoraan tallennusverkkoon. Tämä mahdollistaa datan varmuuskopioinnin ilman palvelimen tai lähiverkon kuormittamista (server-free/LAN-free backup). [3, s. 235] Tallennusverkon kautta nauhakirjasto voidaan jakaa usean palvelimen kesken samalla tavalla kuin levyjärjestelmätkin. Nauhakirjaston jakaminen palvelimien kesken voidaan tehdä joko staattisesti tai dynaamisesti. Staattisessa jakamisessa nauhakirjasto jaetaan useaan virtuaaliseen nauhakirjastoon ja jokaiselle palvelimelle osoitetaan oma virtuaalinen nauhakirjasto (kuva 2.12 vasemmalla). Jokainen nauha-asema ja nauha nauhakirjastossa osoitetaan yksiselitteisesti tiettyyn virtuaaliseen nauhakirjastoon. Virtuaaliset nauhakirjastot vuorottelevat robotin käytöstä, joka vastaa nauhojen siirtämisestä säilytyslokeroiden ja nauha-asemien välillä. Muutosten tekeminen staattiseen nauhakirjaston jakamiseen on aikaa vievää ja kallista, mikä tekee siitä epäkäytännöllistä. [1, s. 201]



Kuva 2.12. Staattinen (vas.) ja dynaaminen (oik.) nauhakirjaston jako [1, s. 201-202]

Dynaaminen nauhakirjaston jakaminen on joustavampaa kuin staattinen jakaminen. Tässä tavassa palvelimet neuvottelevat keskenään nauhakirjaston asemien ja nauhojen käytöstä. (kuva 2.12 oikealla) Yksi palvelimista toimii ”kirjastonhoitajana” (library master) ja muut sen asiakkaina (client). Kirjastonhoitaja koordinoi nauhakirjaston käyttöä. Asiakkaiden on aina ensin pyydettävä lupaa kirjastonhoitajalta, mikäli ne haluavat käyttää nauhakirjastoa. Kirjastonhoitaja varaa nauhan ja aseman ja ilmoittaa asiakkaalle, että se voi nyt käyttää nauhakirjastoa. Asiakas voi tämän jälkeen kirjoittaa suoraan nauhoille tallennusverkon kautta. Dynaaminen nauhakirjaston käyttö on ylläpidollisesti paljon helpompaa, mutta se vaatii enemmän älykkyyttä palvelimilta, mikä ei kaikissa tapauksissa ole mahdollista. [1, s. 201-202]

2.4.4 Portit ja kaapelit

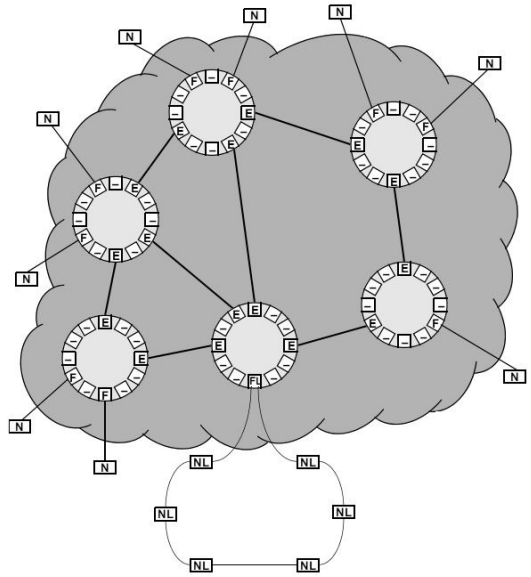
Kuituportit ja -kaapelit yhdistävät tallennusverkon laitteet toisiinsa fyysisellä tasolla. Riippuen tallennusverkon topologiasta kuituportit toimivat eri tavoilla [3, s. 44]. Eri topologiat esitellään tarkemmin luvussa 2.6. Porttien välillä voidaan käyttää erilaisia kuitukaapeleita, joilla saadaan siirrettyä dataa eri etäisyyksien päähän.

Portit

Laitteilla, jotka halutaan liittää tallennusverkkoon, tulee olla siihen soveltuva portti. Palvelimissa tämä tarkoittaa yleensä kuitukortin (HBA) lisäämistä palvelimen laajennusporttiin. Kuituportissa on aina kaksi kanavaa, joista toista käytetään liikenteen lähettämiseen ja toista vastaanottamiseen. Kanavista käytetään yhteisnimitystä linkki. Kuituporteille on määritelty erilaisia toimintatapoja riippuen millaiseen tallennusverkon topologiaan tai laitteeseen ne on kytketty (kuva 2.13). [1, s. 69] Porttien eri toimintatavat on lueteltu alla:

- *N_Portti:* Palvelimet ja levyjärjestelmät ovat verkon päätelaitteita (Node), josta data aina lähtee tai jonne data aina saapuu. Pisteestä-pisteeseen- ja kudostopologiassa palvelimet ja levyjärjestelmät ovat laitteita, joiden portit ovat N_Port-moodissa. [1, s. 70; 3, s. 44]
- *F_Portti:* Kytkimen avulla voidaan muodostaa kudostopologia (Fabric). Verkon solmut (N_Portti) kytkeytyvät kudokseen aina kytkimen F_Portin kautta. [8, s. 18]
- *L_Portti:* Keskittimen (Hub) avulla voidaan muodostaa rengastopologia (Loop). Rengastopologiassa käytetään eri protokollaa kuin kudostopologiassa. L_Portti-moodissa portti voi osallistua renkaaseen. [1, s. 70] Keskittimen portit toimivat tässä moodissa [3, s. 46]
- *NL_Portti:* Laitteet, joiden portti toimii NL_Portti-moodissa (Node Loop), voivat keskustella renkaaseen liitetyn kytkimen FL_Portin kanssa. [3, s. 44]

- *FL_Portti:* Keskittimen muodostama rengastopologia voidaan liittää kudokseen FL_Portin kautta [3, s. 46]. FL_Portti ei voi kommunikoida minkään muun kuin NL_Port:n kanssa [8, s. 18].
- *E_Portti:* Tallennusverkon kytkimiä voidaan yhdistää toisiinsa E_Porttien avulla. E_Portti voi kommunikoida ainoastaan toisen E_Portin tai B_Portin kanssa. [8, s. 18] Kytkimien välisiä linkkejä kutsutaan ISL-linkeiksi, joiden kautta kytkimet siirtävät dataa sekä kudoksen tietoja. [1, s. *70*]
- *B_Portti:* Tallennusverkon siltalaitteet (Bridge) toimivat B_Portti-moodissa. B_Portin kautta kytkin ja silta voivat kommunikoida keskenään. [8, s. 18].
- *G_Portti ja GL_Portti:* Kytkimen portit, jotka voi toimia sekä E_Porttina tai F_Porttina ovat G_Portteja. G_Portti määrittelee portin alustuksen yhteydessä missä moodissa sen pitää toimia. GL_Portti osaa toimia myös FL_Porttina [8, s. 18]

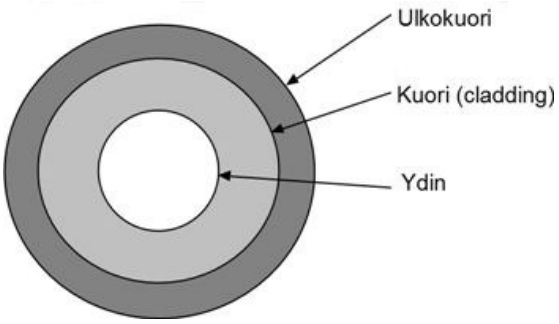


Kuva 2.13. Eri Fibre Channel -porttien liittyminen toisiinsa [8]

Kaapelit

Fibre Channel tukee sekä kuparisia että optisia kaapeleita. Kupariset kaapelit ovat nykyään todella harvinaisia, koska optinen kaapeli tarjoaa huomattavasti paremmat ominaisuudet kupariin verrattuna. Näitä kaapelityyppejä voi myös sekoittaa keskenään samaan kudokseen. Se ei kuitenkaan ole suotavaa.

Kuitukaapelit voidaan luokitella niiden lasisen ytimen paksuuden perusteella. Tallennusverkoissa käytetään kolmea erityyppistä valokuitua, joiden ytimen paksuudet ovat 9, 50 ja 62,5 µm. Valokuidun ydintä ympäröivän kuoren (cladding) paksuus kaikissa kolmessa eri kaapelityypissä on 125 µm. Kuoren tarkoitus on estää valon heijastuminen pois ytimestä (kuva 2.14). Valon pitää kokonaisheijastua mahdollisimman tehokkaasti ytimen reunoista, jotta signaalin taso pysyy hyvänä.

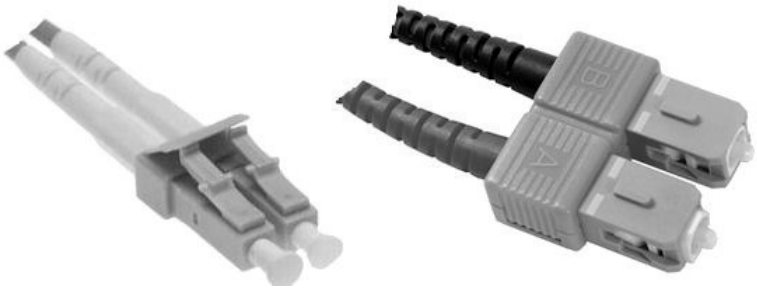


Kuva 2.14. Valokuidun poikkileikkaus

Yhdeksän mikrometrin valokuitukaapeli on ns. yksimuotokuitu (single-mode), jossa käytetään pitkää valon aallonpituutta signaalin lähettämiseen. Yksimuotokuidun maksimipituus on nykyään kymmeniä kilometrejä, joten sitä käytetään lähinnä tilanteissa, joissa data pitää siirtää todella kauas.

50 ja 62,5 mikrometrin valokuitukaapelit ovat monimuotokuituja, jossa käytetään lyhyttä laservalon aallonpituutta. Tämä lyhytaaltoinen laservalo koostuu useista sadoista erimuotoisista laservaloista, jotka heijastuvat eri kulmissa kuidun ytimessä. Tästä johtuen, monimuotokuiduilla kaapelin maksimipituus on paljon pienempi kuin yksimuotokuidulla. 50 mikrometrin valokuitukaapelin maksimipituus on 500 metriä ja 62,5 mikrometrin kaapelissa ainoastaan 175 metriä. [2, s. 44]

Nykyään käytetään kahta erilaista liitintä valokuitukaapelin päissä. SC-liitin on vanhempi ja isompi näistä kahdesta. LC-liitin on uudempi ja pienempi liitin, joka tukee kahden, neljän ja kahdeksan Gbit/s siirtonopeutta. Pienuutensa takia se vie vähemmän tilaa laitteissa ja siksi se on lähes kokokaan syrjäyttänyt SC-liittimen. Pienemmän liittimen ansioista LC-portteja saadaan mahtumaan esimerkiksi kytkimeen huomattavasti tiheämmin kuin SC-portteja (kuva 2.15). [3, s. 43]



Kuva 2.15. Valokuidun LC- (vas.) ja SC-liitin (oik.)

2.5 Protokollat

Tässä luvussa esitellään tallennusverkoissa käytettävät protokollat SCSI sekä Fibre Channel.

2.5.1 SCSI

SCSI-protokolla kehitettiin alun perin tarjoamaan tehokkaan siirtotavan tietokoneiden ja lisälaitteiden välille (kiintolevyt, printterit, skannerit). Siitä on kuitenkin kehittynyt palvelimien I/O-väylässä eniten käytetty standardi. Se on määritelty ensimmäisen kerran jo vuonna 1986. Sen jälkeen siitä on julkaistu useita eri versioita, joista jokainen on kasvattanut SCSI-väylän siirtonopeutta (taulukko 2.1). [2, s. 27]

Taulukko 2.1. Eri SCSI-versioiden siirtonopeus, väylänleveys ja maksimilaitemäärä yhdessä väylässä

SCSI-versio	Siirtonopeus MB/s	Väylänleveys (bittiä)	maksimi laitemäärä
SCSI-2	5	8	8
Wide Ultra SCSI	40	16	16
Wide Ultra SCSI	40	16	8
Wide Ultra SCSI	40	16	4
Ultra2 SCSI	40	8	8
Wide Ultra2 SCSI	80	16	16
Ultra2 SCSI	160	16	16
Ultra320 SCSI	320	16	16

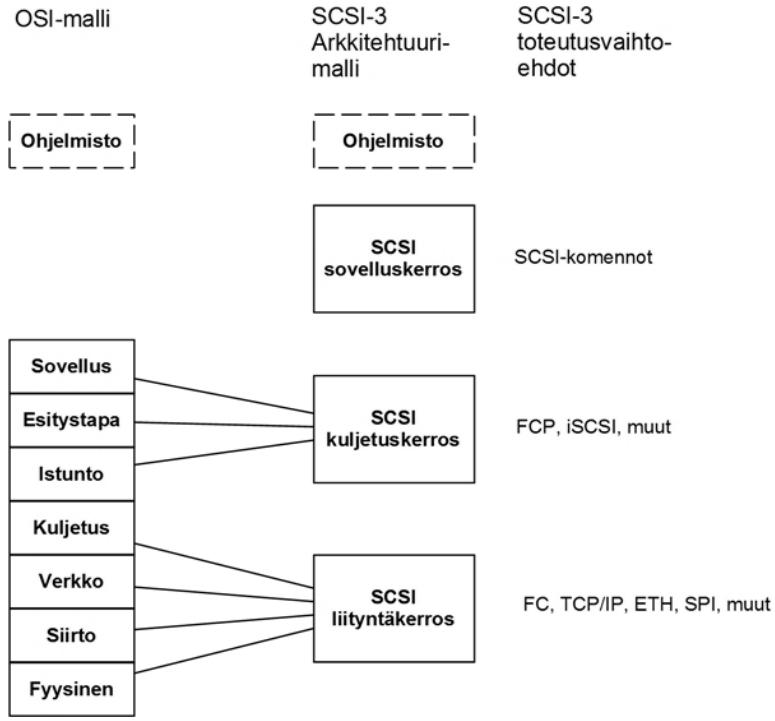
SCSI-protokolla vastaa ainoastaan siitä, että kirjoitus- ja lukuoperaatiot onnistuvat ja että tämä tieto välittyy käyttöjärjestelmälle. Sen ei tarvitse tietää millainen fyysinen tai looginen laite väylän toisessa päässä on. SCSI-protokolla ei myöskään vastaa datalohkojen järjestämisestä sovelluskerrokselle tai siitä, miten data sijoitetaan kiintolevylle. Se tietää ainoastaan, että tietty määrä dataa pitää kirjoittaa tiettyyn paikkaan. [2, s. 27]

SCSI-standardit

SCSI-standardin kaksi ensimmäistä versiota, SCSI-1 ja SCSI-2, määrittelevät lohkotason protokollan, sähköisen ja fyysisen rajapinnan samassa standardissa. SCSI-3 versiossa nämä kolme on eritelty omiin spesifikaatioihin. [4, s. 13]. Tämän jälkeen SCSI-3:sta on julkaistu lukuisia versioita, joista tärkein on vuonna 1995 julkaistu SCSI-3 arkkitehtuurimallin standardi (SAM). SAMissa SCSI-protokolla jaetaan kolmeen eri kerrokseen: SCSI-sovelluskerros, SCSI-kuljetusprotokollat ja SCSI-liityntäkerros (interconnects).

SCSI-sovelluskerros määrittelee komennot joita SCSI-laitteet käyttävät tiedonsiirrossa. Kuljetusprotokollat määrittelevät säännöt ja komennot, joiden avulla SCSI-laitteet kommunikoivat keskenään. SCSI-liityntäkerros sisältää alemman tason protokollat, joiden avulla tieto siirretään laitteiden välillä. Näitä kolmea kerrosta on vaikea sitoa tiettyihin OSI-mallin protokollapinon kerroksiin, koska kahdella alimmalla tasolla voidaan käyttää useaa eri protokollaa, jotka sijoittuvat eri tavalla OSI-mallin protokollapinoon (kuva 2.16). SCSI:n kuljetusprotokollana voidaan käyttää esimerkiksi

FC-, FCoE tai iSCSI-protokollaa. Liityntäkerroksella voidaan käyttää FC-protokollaa, TCP/IP:tä, Ethernetiä tai SCSI-prokollan omaa rinnakkaisväyläliitäntää SPI:tä. Mikäli käytössä on SPI, sovelluskerroksen komennot välitetään suoraan SPI:lle ilman kuljetusprotokollaa. [4, s. 22]



Kuva 2.16. SCSI arkkitehtuurin eri kerrokset OSI-mallissa [4, s. 22]

SCSI-väylä

Yhteen SCSI-väylään voidaan liittää kahdeksasta kuuteentoista laitetta riippuen SCSI-standardista (kuva 2.17). Väylän topologia on ketju (daisy chain), jossa ainoastaan yksi laite voi kommunikoida samanaikaisesti (half-duplex). Jokaisella väylän laitteella tulee olla yksiselitteinen tunniste, myös itse palvelimella. Laitteen tunniste on myös sen prioriteetti SCSI-väylällä. Koska SCSI-standardin viimeiset versiot tukevat kahdeksan laitteen sijaan kuuttatoista laitetta, tunnisteiden prioriteetit ovat menneet hieman sekavaksi. Tunnisteilla 0-7 on korkeampi prioriteetti kuin tunnisteilla 8-15, koska jälkimmäiset on lisätty standardiin jälkeenpäin. Tunnisteiden prioriteetti korkeimmasta aloittaen on 7, 6, 5, 4, 3, 2, 1, 0, 15, 14, 13, 12, 11, 10, 9, 8.

Kuva 2.17. SCSI-väylän ketjutopologia

Kaikkien SCSI-väylään liitetyt laitteet joutuvat varaaman väylän käyttöönsä aina ennen kuin ne voivat alkaa kommunikoida sen kautta. SPI-standardissa on määritelty kaksi tapaa, jolla varaus voidaan suorittaa. Normaalissa valinnassa (normal arbitration) laite, joka haluaa käyttää väylää ja jolla on korkein prioriteetti, voittaa aina väylän käyttöönsä. Muut laitteet jäävät odottamaan seuraavaa valintaa. Tilanteissa, joissa väylä on kuormitettu, matalan prioriteetin omaavat laitteet voivat ”nälkiintyä”, koska ne eivät välttämättä saa väylää koskaan käyttöönsä. SCSI-väylän jakamista ei voida siis pitää reiluna. Valintamenettelyyn on lisätty tämän takia reiluusalgoritmi, jossa jokainen laite pitää kirjaa hävityistä valintayrityksistä. Korkeamman prioriteetin laitteet häviävät valinnassa laitteille, jotka ovat aikaisemmilla kerroilla hävinneet. Näin voidaan estää matalan prioriteetin laitteiden nälkiintyminen. Toinen tapa on ”nopea sovittelu ja valinta” (Quick Arbitration and Selection), jossa tätä tekniikkaa tukevat laitteet voivat ottaa väylän käyttöönsä toisiltaan ilman että väylää erikseen vapautetaan. Mikäli kaikki laitteet eivät tue QAS:ia, aloittaja voi pakottaa väylän vapaaksi, jotta muut laitteet eivät nälkiinny. [1, s. 64-65; 4, s. 129-130]

SCSI-laitteet toimivat isäntä/orja-periaatteella, jossa kommunikoivista laitteista toinen toimii ns. aloittajana (initiotor) ja toinen kohteena (target). Aloittajana yleensä toimii palvelimen SCSI-ohjain ja kohteena jokin tallennuslaite. Aloittajaa aloittaa keskustelun laitteiden välillä lähettämällä komennon kohteelle ja jää odottamaan vastausta. SCSI-protokollaa voidaan pitää siis komento/vastaus-protokollana (command/response). [4, s. 14-15] Aloittaja tunnistaa kohteen kolmen tunnisteen avulla, jotka ovat väylä, kohteen SCSI tunniste ja LUN. Jokainen palvelimen SCSI-liityntä muodostaa oman väylän. Tietyssä väylässä kohteella on oma SCSI-tunniste, jolla identifioidaan tietty fyysinen laite. Viimeiseksi LUN:n avulla fyysisestä laitteesta voidaan tunnistaa looginen levy. [4, s. 14]

SCSI tallennusverkoissa

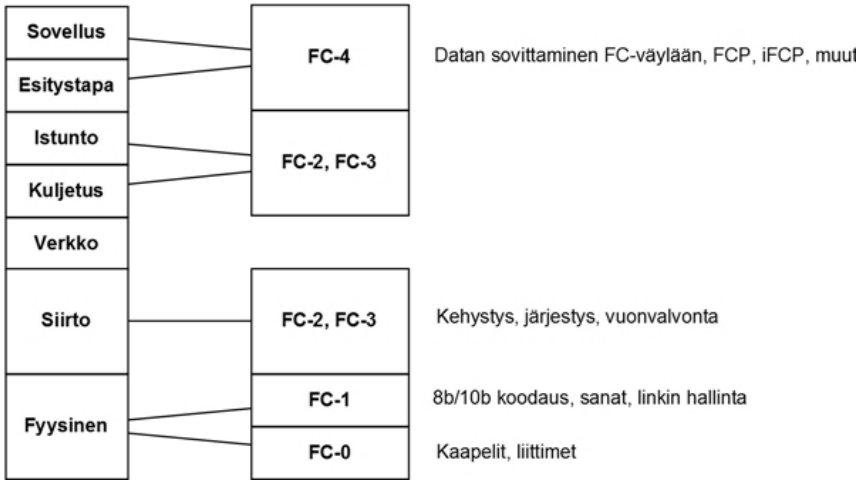
Pelkän SCSI-protokollan avulla on vaikea muodostaa tallennusverkkoa. SCSI-väylään voi liittää enimmillään 16 laitetta, joista käytännössä vain yksi voi olla palvelin. Teoriassa palvelimia voidaan liittää useampi kuin yksi, mutta käytännössä ne eivät toimi hyvin tällä tavalla. SCSI-väylässä ainoastaan yksi voi kommunikoida samanaikaisesti, mikä tekee siitä huonon ratkaisun järjestelmiin, joissa liikennemäärät ovat suuria. Klusteritekniikka on mahdollista SCSI-väylän avulla, kun ketjun molempiin päihin sijoitetaan palvelin. Toinen palvelimista on aktiivinen ja toinen valmiustilassa, jos toinen hajoaa. SCSI:lla ei saavuteta kuormaa jakavia klustereita. SCSI-kaapelien pituusrajoitus asettaa myös omat rajoitteensa tallennusverkon muodostamiselle.

Edellä mainituista syistä pelkkää SCSI-protokollaa ei voida käyttää tallennusverkon muodostamiseen. SCSI:n jaettu arkkitehtuurimalli (SAM) mahdollistaa muiden protokollien käyttämisen kuljetus- ja liityntäkerroksilla. Fibre Channel -protokollan (tai muun vastaavan) avulla SCSI-väylä voidaan korvata verkolla. SCSI-protokollan komentoja käytetään edelleen kommunikoinnissa tämän verkon yli. Tällä tavalla

tallennusverkko pysyy näkymättömänä sovelluksille ja käyttöjärjestelmän ylemmille kerroksille. Tallennusverkon tallennuslaitteiden on myös kommunikoitava SCSI-protokollan avulla. [1, s. 65]

2.5.2 Fibre Channel

Fibre Channel on tällä hetkellä kaikista käytetyin verkkoteknologia tallennusverkoissa, ja sen on määritellyt ANSI T11 -komitea. Protokollaa alettiin työstää jo vuonna 1988 ja sen ensimmäinen versio hyväksyttiin standardiksi vuonna 1994. Se suunniteltiin alun perin runkoverkkoteknologiaksi yhdistämään lähiverkkoja, mutta sen ominaisuudet ovat osoittautuneet hyviksi myös tallennusverkoissa. Fibre Channel suunniteltiin tarjoamaan korkeaa siirtonopeutta, matalaa viivettä, korkeaa luotettavuutta ja skaalautuvuutta, jotka ovat myös vaatimukset tallennusverkolle. Fibre Channelin päällä pystytään kuljettamaan monia eri protokollia, myös IP-protokollaa. Fibre Channel protokollapino voidaan jakaa viiteen osaan: FC-0 – FC-4 (kuva 2.18). Näistä FC-0 – FC-3 sisältävät keskeiset kommunikointitekniikat kuten fyysisen kerroksen ja datan kuljetuksen määritykset sekä osoitteistuksen. Ylin taso FC-4 määrittelee, miten ylemmän kerroksen protokollat sovitetaan Fibre Channeliin. Tasojen rinnalla toimii myös palveluita, jotka ovat välttämättömiä datan kuljettamiseen ja tallennusverkon hallintaan. [1, s. 66–67; 4, s. 84-85]



Kuva 2.18. Fibre Channel --protokollan kerrokset OSI-mallissa

FC-0

Fibre Channelin alin kerros FC-0 määrittelee fyysisen siirtokanavan ominaisuudet (kaapelit ja liittimet). Fibre Channelissä data siirretään sarjamuodossa, ei rinnakkaisväylää pitkin niin kuin esimerkiksi SCSI-väylässä. Sarjamuotoinen siirtotapa mahdollistaa suuremman siirtonopeuden, koska siinä ei esiinny rinnakkaisväylässä olevaa hajontaongelmaa. Sarjamuotoinen siirtotapa mahdollistaa myös paljon pitemmät siirtovälit.

Fibre Channel tukee monia eri siirtonopeuksia. Siirtonopeuksia on pyritty kasvattamaan muutaman vuoden välein, koska siirrettävän datan määrä kasvaa joka vuosi. Siirtonopeudet voidaan jakaa kahteen ryhmään (taulukko 2.2), joista toinen perustuu kahden potenssiin (Base2) ja toinen kymmenen moninkertoihin (Base10). [1, s. 71; 10, s. 37]

Taulukko 2.2. Fibre Channel -protokollan eri versioiden siirtonopeus, määrittelyn valmistumisvuosi ja markkinoilletulovuosi

FC-versio	Siirtonopeus (MB/s)	Määrittely valmistunut	Markkinoille
1GFC	100	1996	1997
2GFC	200	2000	2001
4GFC	400	2003	2005
8GFC	800	2006	2008
16GFC	1600	2009	2011
32GFC	3200	2012	Kysynnän mukaan
FC-versio	**Siirtonopeus (MB/s)**	**Määrittely valmistunut**	**Markkinoille**
10GFC	1200	2003	2004
20GFC	2400	2008	2008
40GFC	4800	2009	2011
80GFC	9600	Tulevaisuudessa	Kysynnän mukaan

Fibre Channel -standardi vaatii, että siirtokanavan bittivirhesuhteen (BER) tulee olla vähintään 10^{-12} eli jokaista siirrettyä terabittiä kohden sallitaan ainoastaan yksi bittivirhe. Bittivirheiden havaitseminen on toteutettu protokollan ylemmillä kerroksilla. Kuituverkon herkin lähde bittivirheille on itse valokuidut. Liian tiukka kulma kaapelissa tai pieni fyysinen vaurio saattaa aiheuttaa sen, että em. bittivirhesuhteeseen ei päästä. [1, s. 72]

FC-1

FC-1 määrittelee mitä datalle pitää tehdä ennen kuin se siirretään kuitukaapelia pitkin. Siinä määritellään myös linkin hallintaan käytettävät ”sanat” (ordered sets). Digitaalisissa siirtotekniikoissa lähettäjän ja vastaanottajan on aina synkronoitava kellopulssinsa, jotta vastaanottaja tietää, millä hetkellä sen pitää lukea dataa signaalista. Sarjakaapelissa kellopulssi on lähetettävä samaa kaapelia pitkin kun datakin, joten vastaanottajan on synkronoitava kellonsa datasignaalista.

Fibre Channel käyttää datan siirtämisessä binäärikoodausta. Binäärikoodauksen ongelmana on ns. ”jitter”, jolla tarkoitetaan kanavan fyysisistä ominaisuuksista johtuvaa signaalin pidentymistä siirtokanavassa. Pidentyminen voi johtaa siihen, että dataa luetaan väärästä kohdasta signaalia. Tämä muodostuu ongelmaksi, jos datassa on monta peräkkäistä nollaa tai ykköstä. Fibre Channel käyttää ongelman kiertämiseen *8b/10b*

koodausta (8b/10b encoding), jossa jokainen kahdeksan bitin sarja muunnetaan kymmeneksi bitiksi, ennen kuin se siirretään siirtokanavalle. Tämä tietysti vähentää siirtokanavan hyötynopeutta, mutta siitä saatavat edut merkitsevät enemmän. 8b/10 koodauksesta saatavat hyödyt ovat:

- Koodauksessa käytetään ainoastaan niitä kymmenen bitin sarjoja, joissa nollia ja ykkösiä ei ole yli viittä kappaletta peräkkäin. Näin signaali vaihtuu vähintään viiden bitin välein.
- 8b/10b koodaus sisältää tasaisen jakauman nollia ja ykkösiä. Fibre Channeliä käsittelevistä laitteista saadaan näin yksinkertaisempia ja halvempia.
- Yli jäävät kymmenen bitin sarjat, jotka eivät esitä kahdeksan bitin tavuja, voidaan käyttää linkin hallintaan. [1, s. 74–75]

Fibre Channel muodostaa neljästä kymmenen bitin sarjasta yhden 40 bittiä sisältävän sanan. Nämä sanat jaetaan datasanoiksi (transmission word) ja järjestetyiksi joukoiksi (ordered sets). Datasanat sisältävät neljä tavua dataa. Järjestettyjä joukkoja käytetään linkin hallintaan. FC-1 määrittelee useita järjestettyjä joukkoja, joita käytetään linkkitason alustukseen ja hallintaan. Datasanat voivat sijaita ainoastaan kehyksen alkumerkin (SOF) ja loppumerkin välissä (EOF). Järjestetyt joukot sijaitsevat päinvastoin loppu ja alkumerkin välissä. [1, s. 75]

FC-2

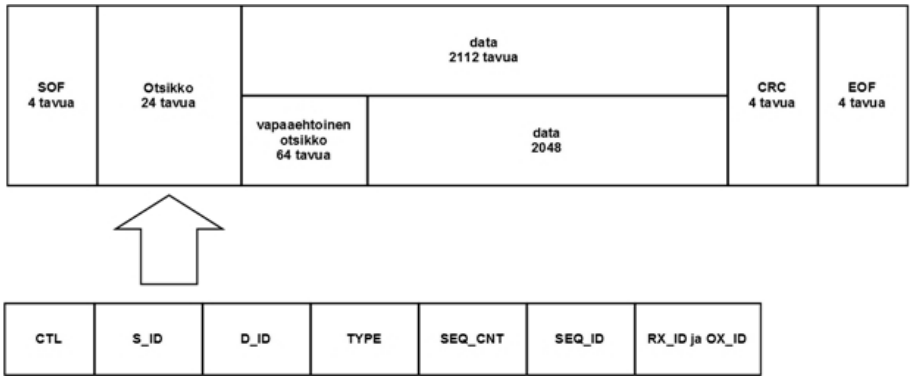
FC-2 kerroksen kehystys ja signalointi takaavat luotettavan siirtoyhteyden. Se myös määrittelee datan siirtomekanismit. Tallennusverkon topologia tunnistetaan automaattisesti FC-2 kerroksella. FC-2 tukee sekä pisteestä-pisteeseen- (point-to-point), rengas- (arbritrated loop) ja kytkettyä kudostopologiaa. (switched fabric). [10, s. 37]

FC-2 käyttää kolmekerroksista hierarkiaa tiedonsiirrossa. Hierarkian korkeimmalla tasolla ovat ns. loogiset kanavat (exchange), joita FC-2 muodostaa kahden päätelaitteen välille, esimerkiksi palvelimen ja levyjärjestelmän välille (N_Porttien välille). Päätelaitteiden välillä voi olla useita loogisia kanavia samanaikaisesti ja ne voivat olla joko yksi- tai kaksisuuntaisia (unidirectional/bidirectional). Loogiset kanavat mahdollistavat saapuvan datan nopean välittämisen oikealle vastaanottajalle korkeammalle protokollakerrokselle.

Hierarkian toisella tasolla ovat sekvenssit (sequence). Sekvenssit ovat suurempia datayksiköitä, jotka siirretään lähettäjän ja vastaanottajan välillä. FC-2 takaa, että sekvenssit saapuvat oikeassa järjestyksessä vastaanottajan korkeammalle protokollakerrokselle. Jokainen sekvenssi sisältää uniikin järjestysnumeron, jonka avulla ne voidaan järjestää oikeaan järjestykseen. Järjestysnumeron avulla voidaan myös ilmoittaa lähettäjälle virheellisistä sekvensseistä.

Hierarkian alin taso koostuu kehyksistä. Kehykset voidaan jakaa hallinta- ja datakehyksiin. Datakehykset voivat kuljettaa 2112 tavua hyötykuormaa yhdessä

kehyksessä. Hallintakehykset eivät sisällä hyötykuormaa. Niitä käytetään ainoastaan signalointiin päätelaitteiden välillä. Mikäli sekvenssi on pitempi kuin 2112 tavua, se pilkotaan useaan kehykseen. [1, s. 77; 11]



Kuva 2.19. Fibre Channel -protokollan kehys

Fibre Channel kehys (kuva 2.19) alkaa aina kehyksen neljän tavun alkumerkillä (SOF) ja päättyy neljän tavun loppumerkkiin (EOF). Näiden välissä ovat 24 tavun otsikko, 2112 tavun datakenttä sekä neljän tavun CRC-kenttä. Otsikko koostuu lähde- (S_ID) ja kohdeosoitteesta (D_ID), tyyppikentästä (TYPE), kehysten määrästä sekvenssissä (SEQ_CNT), kehyksen tunnisteesta (SEQ_ID) sekä loogisen kanavan tunnisteista (OX_ID ja RX_ID). Tämän lisäksi otsikko sisältää kehyksen käsittelyyn liittyviä kenttiä (CTL kentät). Datakenttään voidaan lisätä 64 tavun mittainen valinnainen otsikko, loput datakentästä täytetään siirrettävällä datalla (2048 tavua). CRC-kentän avulla vastaanottaja voi määritellä tapahtuiko tiedon siirron aikana virheitä ja saapuiko kehys ehjänä perille. Virheelliset kehykset lähetetään uudelleen. [1, s. 77; 11]

FC-2 tasolla on määritelty myös vuonvalvonta (flow control) ja eri palvelutasot (CoS). Fibre Channel tukee sekä päästä-päähän-tapahtuvaa että linkkien välistä vuonvalvontaa. Vuonvalvonnan tarkoitus on estää lähettäjää lähettämästä sekvenssejä liian suurella nopeudella, jolla vastaanottaja ei pysty niitä käsittelemään. Päästä-päähän-vuonvalvonnassa N_Portit neuvottelevat keskenään lähetysnopeudesta. Linkkien välisessä vuonvalvonnassa, kudoksen jokainen linkki neuvottelee erikseen sopivan lähetysnopeuden. Fibre Channel tukee myös kuutta eri palvelutasoa, joilla voidaan vaikuttaa kehysten reititykseen ja kuittaukseen. [1, s. 78; 11]

FC-3

FC-3 taso on ollut kehitystyön alla jo vuodesta 1988. Tällä hetkellä standardi ei tarjoa mitään palveluita tällä tasolla. Tulevaisuudessa tämä taso saattaa tarjota lisäpalveluita, kuten usean kanavan tuki (multipath), datan pakkaus ja salaus, ryhmälähetys (multicast), datan peilaus ja muita RAID-tasoja.

Vaikka näitä palveluita ei ole lisätty standardiin vielä, useat valmistajat ovat kuitenkin toteuttaneet niitä tuotteisiinsa. Usean kanavan tuki saadaan käyttöön sopivalla

sovelluksella palvelimissa. Jotkut kytkimet tukevat myös moninkertaisia kytkimien välisiä linkkejä (ISL trunking). [1, s. 82; 10, s. 38; 11]

FC-4

Fibre Channelin korkein taso FC-4 toimii sovellusrajapintana (API) sovelluskerroksen protokollille. Sen tehtävä on sovittaa sille annettu data kuitukanavaan. Fibre Channel pystyy kuljettamaan useita eri sovelluskerroksen protokollia, esimerkiksi SCSI, IP ja ATM. [11]

SCSI:n sovellusprotokollasta käytetään nimitystä Fibre Channel Protocol (FCP). Sen tehtävä on sovittaa SCSI-komennot Fibre Channelin alemmille kerroksille. FCP-protokollan idea on korvata SCSI-väylä ja ketjutopologia tallennusverkolla ja piilottaa muutos käyttöjärjestelmältä. FCP:tä tukevan uuden ajurin asentamisella saadaan käyttöjärjestelmä näkemään tallennusverkosta näytetyt levyt normaaleina SCSI-laitteina. Tämä yksinkertaistaa ja helpottaa siirtymistä puhtaista SCSI-laitteista tallennusverkon käyttöön. FCP-laiteajurin pitää myös osata muuttaa SCSI-väylän rinnakkaisrakenne Fibre Channelin sarjamuotoiseksi liikenteeksi. [1, s. 87–88]

Liittyminen kudokseen

Fibre Channel käyttää kolmivaiheista sisäänkirjautumismekanismia laitteiden liittämiseen kudokseen. Näistä vaiheista käytetään nimityksiä kudos-, portti- ja prosessisisäänkirjautuminen (fabric login, port login, process login).

Kudokseen sisäänkirjautuminen (FLOGI) tapahtuu N_Portin tai NL_Portin ja F_Portin välillä heti kun niiden välille on muodostettu fyysinen yhteys. Ennen sitä mitään muita kehyksiä ei voida vaihtaa laitteiden välillä. Portit sopivat tiedonsiirrossa käytettävistä parametreista ja F_Portti lähettää siihen liitetyllä laitteen dynaamisen porttiosoitteen, josta puhutaan lisää seuraavassa kohdassa.

Porttisisäänkirjautuminen (PLOGI) suoritetaan kahden N_Portin välillä. Ne muodostavat välilleen istunnon (session), jossa ne vaihtavat palveluparametreja ja tekevät itsensä tunnetuksi toisilleen. Tämän jälkeen portit voivat vaihtaa sovelluskerroksen (FC-4) komentoja keskenään.

Prosessisisäänkirjautuminen (PRLI) muodostaa istunnon kahden sovellustason prosessin välille sen jälkeen kun PLOGI on suoritettu. PRLI on valinnainen proseduuri FC-2 kerroksen näkökulmasta, mutta jotkin sovellustason protokollat vaativat sitä ennen kuin ne voivat siirtää dataa keskenään. [1, s. 82-84; 10, s. 78-79]

Osoitteistus

Fibre Channel-verkossa kaikilla laitteilla pitää olla yksilöllinen tunniste, jotta niihin voidaan viitata yksiselitteisesti. Fibre Channelissä laitteen osoitetta kutsutaan World Wide Name:ksi (WWN). Se on maailmanlaajuisesti uniikki osoite, jonka laitteen valmistaja on määrännyt kyseiselle laitteelle. WWN on 64 bittiä pitkä

heksadesimaaliluku. Uudempi WWN:n rakenne alkaa yleensä heksadesimaalinumerolla 5 tai 6, jota seuraa kuusi numeroa pitkä valmistajan tunniste. Osoitteen loppuosa identifioi kyseisen laitteen. WWN:t jaetaan WWNN:iin (Wolrd Wide Node Name) ja WWPN:iin (World Wide Port Name). WWNN on osoite koko laitteelle ja WWPN on osoite tietylle portille. Laitteella on siis yksi WWNN, joka identifioi koko laitteen. Lisäksi jokaisella laitteen FC-portilla on oma WWPN. [1, s. 84; 10, s. 58-59]

64-bittisen osoitteen käyttäminen reitityksessä saattaa aiheuttaa viivettä. Siksi kudoksessa jokaiselle WWPN-osoitteelle määrätään fabric loginin aikana 24-bittinen porttiosoite, jota käytetään kudoksessa lähettäjän ja vastaanottajan identifioimiseksi. Porttiosoite ei ole laiteriippuvainen vaan kudos luo sen automaattisesti. Porttiosoite koostuu kolmesta kahdeksan bitin kentästä. Eniten merkitsevät bitit 23-16 identifioivat kytkimen toimialueen (domain), 15-8 alueen (area) ja 7-0 portin. Osa domain-osoitteista on varattu, joten yhdessä kudoksessa voi siis olla 239 kytkintä, joissa jokaisessa voi olla 256 aluetta, joissa kaikissa voi olla 256 porttia (15,663,104 osoitetta). Kudoksen nimipalvelin vastaa WWN-osoitteiden ja porttiosoitteiden sitomisesta toisiinsa, johon tieto tallentuu fabric loginin aikana. [10, s. 61]

Rengastopologiassa porteille määrätään myös 24-bittinen osoite, josta ensimmäiset 16-bittiä identifioivat renkaan. Rengas täytyy identifioida, jos se on yhteydessä kytkimeen FL-portin kautta. Jos rengas ei ole yhteydessä kytkimeen, ensimmäiset 16 bittiä osoitteesta ovat nollia. Loput kahdeksan bittiä identifioivat laitteen renkaassa. [10, s. 62] Kahdeksalla bitillä on mahdollista esittää 256 osoitetta, mutta 8b/10b koodauksesta johtuen ainoastaan 127 osoitetta on mahdollisia, joissa nollat ja ykköset ovat tasaisesti jakautuneet. Yksi osoitteista pitää jättää mahdolliselle kytkimelle, joten yhteen renkaaseen voi liittää 126 laitetta. [1, s. 85]

2.6 Topologiat

Fibre Channel tukee kolmea eri topologiaa, jotka ovat pisteestä-pisteeseen (point-to-point), rengas (arbritrated loop) sekä kytketty kudos (switched fabric). Pisteestä-pisteeseen-yhteys mahdollistaa full duplex -liikenteen kahden laitteen välillä. Rengas-topologia mahdollistaa kahden tai useamman laitteen välisen kommunikoinnin jaetussa rengasväylässä, jossa laitteet kilpailevat väylän käytöstä. Kytketty kudos mahdollistaa kaikkien laitteiden samanaikaisen kommunikoinnin verkossa. [3, luku 4]

2.6.1 Pisteestä pisteeseen

Pisteestä-pisteeseen-topologia on yksinkertaisin kolmesta mahdollisesta topologiasta. Se on suora yhteys kahden N_Portin välillä (kuva 2.20). Ennen kuin laitteet voivat kommunikoida keskenään, niiden on suoritettava porttisisäänkirjautuminen toistensa kanssa, jossa ne sopivat tiedonsiirrossa käytettävistä parametreista. Koska Fibre

Channel -kaapeleissa on aina erillinen väylä lähetykselle ja vastaanottamiselle, laitteet voivat lähettää ja vastaanottaa täydellä nopeudella samanaikaisesti. [3, luku 4.1]

Kuva 2.20. Pisteestä-pisteeseen-topologia

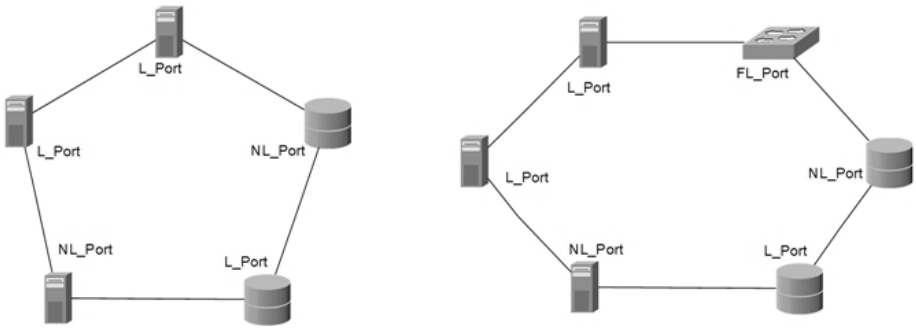
Vaikka topologia näyttääkin samalta kuin ulkoista SCSI-laitetta käyttävä palvelin, on Fibre Channelilla toteutettu pisteestä-pisteeseen topologia huomattavasti parempi vaihtoehto. Fibre Channel -levyjärjestelmästä kiintolevyjen jakaminen palvelimelle on paljon joustavampaa ja palvelin ja levyjärjestelmä voivat olla hyvin kaukana toisistaan verrattuna SCSI-kaapelin 25 metrin pituusrajaan. SCSI-väylä ei myöskään mahdollista full duplex-liikennettä. [1, s. 89]

Pisteestä-pisteeseen-topologian heikkous on sen huono skaalautuvuus. Levyjärjestelmään on mahdollista kytkeä vain niin monta palvelinta kuin siinä on portteja. Yksi portti pystyy yleensä siirtämään useamman kuin yhden palvelimen liikenteen, joten pisteestä-pisteeseen-topologia tuhlaa portteja. Palvelimien määrän lisääntyessä, täytyy myös levyjärjestelmän porttien määrän kasvaa. [3, s. 96-97]

2.6.2 Rengas

Pisteestä-pisteeseen-topologiaa ei voida pitää verkkona, koska se yhdistää palvelin- ja tallennuskerroksen suoraan toisiinsa. Verkon avulla tarvittavien fyysisten porttien määrää voidaan vähentää, koska useamman palvelimen liikenne voidaan kuljettaa samaa kaapelia pitkin. Rengas-topologiassa laitteet muodostavat renkaanmallisen väylän, jonka käytöstä ne kilpailevat. Dataa voidaan kuljettaa renkaassa ainoastaan yhteen suuntaan [1, s. 93].

Rengas-topologiat voidaan jakaa yksityisiin (kuva 2.21 vasemmalla) sekä julkisiin renkaisiin (kuva 2.21 oikealla). Yksityinen rengas on suljettu rengas, jossa laitteiden portit ovat joko L_Portteja tai NL_Portteja. Julkisessa renkaassa yksi tai useampi renkaan laitteista on kytkin, joka mahdollistaa kommunikoinnin renkaan ulkopuolelle. Renkaassa ainoastaan yksi kytkin voi olla aktiivinen kerrallaan. Toiset kytkimet ovat ainoastaan valmiustilassa, jos aktiivinen kytkin hajoaa. Jotta renkaassa oleva laite voi kommunikoida kytkimen FL_Portin kautta, sillä täytyy olla NL_Portti. Pelkkä L_Portti ei osaa toimia kudoksen solmuna kuten NL_Portti. L_Portti osaa kommunikoida ainoastaan renkaissa. [1, s. 94-95]



Kuva 2.21. Yksityinen ja julkinen rengas

Rengas-topologian voi muodostaa käyttämällä tallennusverkon keskitintä. Renkaan kokoa voi helposti kasvattaa lisäämällä useita keskittimiä sarjaan. Vähintään kolmesta keskittimestä voi muodostaa myös renkaan, jolloin yhden linkin hajoaminen keskittimien välillä ei katkaise yhteyttä renkaan tiettyjen osien välillä. Redundanttisuutta voi edelleen kasvattaa lisäämällä kaksi erillistä rengasta. [3, s. 99-104]

Rengastopologian vahvuuksia ovat sen yksinkertaisuus ja halpuus. Tallennusverkko saadaan nopeasti pystytettyä käyttämällä keskitintä ilman suurta suunnittelun tarvetta. Rengas tarjoaa myös parempaa skaalautuvuutta kuin pisteestä-pisteeseen-topologia. Renkaaseen ei kuitenkaan voi liittää kuin 126 laitetta ja jaettu väylä rajoittaa siirtomäärää merkittävästi. Kytkinten halventuminen onkin johtanut nopeasti rengas-topologioiden vähentymiseen.

2.6.3 Kytketty kudos

Kytketty kudos on tällä hetkellä käytetyin ratkaisu tallennusverkoissa. Se tarjoaa kaikkien siihen kytkettyjen laitteiden välisen full duplex-liikenteen lisäksi monipuolisia palveluita saavutettavuuden ja luotettavuuden parantamiseksi. Yhdistämällä kytkimiä toisiinsa saadaan muodostettua laajoja useita satoja laitteita kattavia tallennusverkkoja. Kytkemällä eri tavoin kytkimet toisiinsa voidaan muodostaa erilaisia kudos-topologioita, joista suosituimpia ovat kaskadi-, kehä- (ring, loop of switches), solmuverkko- (mesh) ja Core-Edge-topologia.

Tallennusverkon topologia on suunniteltava huolellisesti, koska jokainen kytkin lisää viivettä noin yhdellä millisekunnilla [3, s. 114]. Topologian pitää olla sellainen, että palvelimien ja levyjärjestelmien välillä on mahdollisimman vähän hyppyjä (kytkimiä), mutta silti verkon pitää olla skaalautuva ja luotettava. Yleinen tapa saatavuuden ja luotettavuuden parantamiseen, on käyttää kahta erillistä kudosta palvelimen ja tallennuslaitteiden välillä. Näin tallennusverkko selviää kokonaisen kudoksen kaatumisestakin. [3, s. 111]

Kaskadi

Yksinkertaisin topologia on kytkeä käytössä olevat kytkimet riviin. Tällaisesti rivistä käytetään nimeä kaskadikytkentä (kuva 2.22). Kahden kytkimen välillä voidaan käyttää useaa ISL-linkkiä, jotta yksi ISL-linkki ei muodostu pullonkaulaksi. [5, s. 148]

Kuva 2.22. Kaskaditopologia

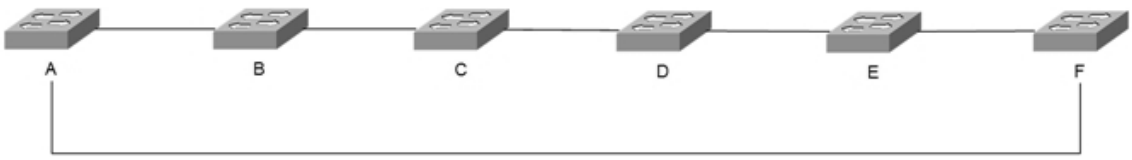
Kaskadikytkentä on helppo muodostaa, mutta se ei ole skaalautuva, suorituskykyinen eikä luotettava. Kun kaskadikytkentään lisätään kytkin jompaan kumpaan päähän, hyppyjen määrä verkon toiselta reunalta toiselle kasvaa, mikä aiheuttaa viivettä tiedonsiirtoon. Koska kaikki liikenne kulkee aina samaa reittiä kytkimien välillä, ISL-linkit saattavat ruuhkautua ja muodostua verkon pullonkaulaksi. Eli kun laitteiden määrä verkossa kasvaa, verkon suorituskyky laskee. Tämä ei ole toivottava piirre tallennusverkossa. [5, s. 148]

Toinen ongelma, joka kaskaditopologiaan liittyy, on jonkin keskellä olevan kytkimen tai ISL-linkin hajoaminen. Tästä seuraa se, että kudos hajoaa kahdeksi saarekkeeksi, joiden välillä ei pysty enää kommunikoimaan. Yksittäisen laitteen vikaantuminen ei saa estää saatavuutta tallennusverkoissa. [5, s. 149]

Edellä mainituista syistä johtuen kaskaditopologian käyttöä tulisi välttää, mikäli se on mahdollista. Se on kuitenkin yksinkertainen topologia, joka on halpa ja helppo toteuttaa, mikä tekee siitä hyvän valinnan pieniin tallennusverkkoihin. Kahden kytkimen muodostamassa kaskadissa, nämä ongelmat eivät pääse esiintymään voimakkaasti. Siksi kaskadia suositellaan käytettäväksi ainoastaan kahden kytkimen muodostamassa kudoksessa tai tilanteissa, joissa liikenne tallennusverkossa pysyy suurilta osin saman kytkimen sisällä. [5, s. 149]

Kehä

Kaskadista saadaan muodostettua kehä (kuva 2.23) yhdistämällä kaskadin päissä olevat kytkimet ISL-linkin avulla [5, s. 149]. Kehä-topologiaa ei pidä sekoittaa keskittimillä muodostettavaan rengastopologiaan. Kehä on kytkimillä muodostettava topologia, joka tarjoaa varareitin, mikäli jokin kytkimistä tai linkeistä katkeaa. Tästä syystä kehä on parempi ratkaisu kuin kaskadi.



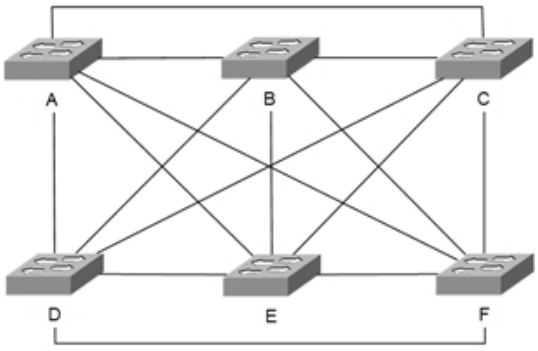
Kuva 2.23. Kytkimien muodostama kehätopologia

Saatavuuden parantumisen lisäksi, kehä on myös suorituskykyisempi kuin kaskadi. Kaskadissa kaikki liikenne kulkee samaa reittiä verkon ääripäiden välillä. Kehätopologiassa liikenne kulkee aina lyhintä reittiä, jos kudoksessa on käytössä FSFP. Esimerkiksi kytkimen A ja F välinen liikenne ei enää vaadi viittä hyppyä vaan ainoastaan yhden hypyn AF takia. Kehällä on keskimäärin lyhyempi hyppymäärä kuin kaskadissa. [5, s. 149]

Vaikka kehätopologia on joissain määrin parempi kuin kaskadi, samat ongelmat koskevat silti molempia. Kytkimien lisääminen topologiaan kasvattaa kehässäkin hyppyjen määrää, mikä pienentää suorituskykyä ja luotettavuutta. Yhdeltä osin kehä on, jopa huonompi ratkaisu kuin kaskadi. Jos kehään halutaan lisätä kytkin, täytyy yksi linkki kehästä rikkoa väliaikaisesti, jotta kytkin voidaan lisätä kehään. Näistä syistä kehätopologiaa ei myöskään kannata käyttää kovin suurissa tallennusverkoissa. [5, s. 149-150]

Solmuverkko

Solmuverkkotopologiaan (Mesh) vaaditaan vähintään neljä kytkintä. Periaate solmuverkoissa on se, että jokaisesta kytkimestä on linkki jokaiseen toiseen kytkimeen, mikä tekee siitä erittäin varman (kuva 2.24). Solmuverkko ratkaisee monia ongelmia, mitä liittyy kaskadi- tai kehätopologian käyttöön. Useankaan ISL-linkin katkeaminen ei välttämättä aiheuta katkosta verkon eri osien välillä, koska kytkimillä on aina useita varareittejä. Täydellisessä solmuverkossa palvelin ja tallennuslaitteen välillä on maksimissaan kaksi kytkintä, joten verkko ei aiheuta merkittävästi viivettä tiedonsiirtoon. Kytkimien lisääminen verkkoon ei myöskään aiheuta katkosta, kuten kehässä. [3, s. 114-116; 5, s. 151]



Kuva 2.24. Täydellinen solmuverkkotopologia

Vaikka solmuverkko ratkaisee monia ongelmia edellisiin topologioihin verrattuna, seuraa siitä myös monia uusia ongelmia. Suuri ISL-linkkien määrä kehittyy solmuverkon ongelmaksi, koska suuri osa kytkimien porteista joudutaan käyttämään ISL-linkkien muodostamiseen. Jos solmuverkossa on neljä 16-porttista kytkintä, joista jokaisen välille muodostetaan ISL, tarvitaan kaikista 64 portista 12 ISL-linkkien muodostamiseen. Mikäli linkit halutaan vielä kahdentaa (trunk), portteja kuluu 24. [3, s.

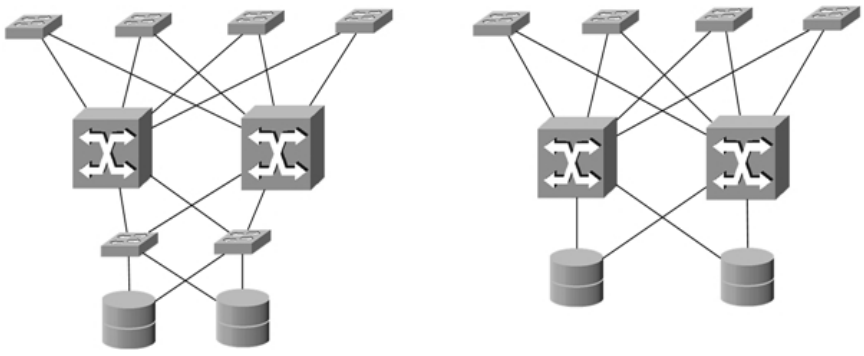
116] Solmuverkko muodostuu erittäin nopeasti epätaloudelliseksi, kun kytkimien määrä verkossa nousee. Kun kytkimien määrä nousee yli puoleen, siitä määrästä mitä yhdessä kytkimessä on portteja, vapaiden porttien määrä alkaa vähentyä. [5, s. 152]

Solmuverkossa kahden kytkimen välillä on ainoastaan yksi ISL. Lisäämällä ISL-linkkien määrää ainoastaan pahentaa skaalautuvuusongelmaa. Solmuverkko ei siis merkittävästi lisää suorituskykyä, koska normaalitilanteessa kaikki liikenne kahden kytkimen välillä kulkee yhtä linkkiä pitkin. [5, s. 153]

Core-Edge

Core-Edge-topologia (CE) on tällä hetkellä käytetyin topologia tallennusverkoissa. Se on alun perin johdettu tähtitopologiasta, jossa yksi kytkin keskellä yhdistää kaikki muut reunoilla olevat kytkimet. CE:ssä kytkimet jaetaan kahteen kerrokseen, joista alemman kerroksen kytkimiä kutsutaan runkokytkimiksi (core) ja ylemmän kerroksen kytkimiä reunakytkimiksi (edge). Ytimessä on kaksi kytkintä, joiden tehtävä on yhdistää reunakytkimet toisiinsa. Runkokytkimet ovat redundanttisia keskenään. Kaikista reunakytkimistä on linkki kumpaakin kytkimeen ytimessä. Runkokytkimien tehtävä on siis yhdistää reunakytkimet toisiinsa. [3, s. 119; 5, s. 154]

CE-topologia on erittäin skaalautuva, suorituskykyinen ja luotettava. Tallennusverkon kokoa voidaan helposti kasvattaa lisäämällä verkkoon uusia reunakytkimiä ja liittämällä ne runkokytkimiin. Runkokytkinten portteja käytetään yleensä ainoastaan ISL ja IFL-linkkien muodostamiseen, jotta portteja ei tuhlata turhaan ytimessä. CE-topologia takaa hyvän suorituskyvyn, koska jokainen reunakytkin on ainoastaan kahden hypyn päässä toisesta reunakytkimestä. Lisäksi jokaisesta kytkimestä on ISL-linkki molempiin runkokytkimiin, joten molempia linkkejä voidaan käyttää aktiivisesti kuormanjakamiseen. Jokaisen kytkimen liikenne kulkee aina vain ja ainoastaan sille tarkoitettuja ISL-linkkejä pitkin, joten CE-topologiassa ei pääse syntymään tilannetta, jossa jokin tietty ISL-linkki kuormittuu muiden kytkimien välisestä liikenteestä. Luotettavuutta CE-topologia lisää redundanttisuuden avulla. CE-topologiassa kaikki kytkimet ja linkit on kahdennettu, joten yhden laitteen tai linkin hajominen ei vaikuta saatavuuteen. [3, s. 119-120; 5, s. 154-155]



Kuva 2.25. Kaksi eri tapaa muodostaa Core-Edge-topologia

Tallennuslaitteet voidaan liittää CE-tpologiaan kahdella eri tavalla. Ensimmäisessä tavassa kaikki laitteet (palvelimet ja tallennuslaitteet) kytketään reunakytkimiin ja runkokytkimiä käytetään ainoastaan ISL- ja IFL-linkkien muodostamiseen (kuva 2.25 vasemmalla). Toinen tapa on kytkeä tallennuslaitteet suoraan runkokytkimiin (kuva 2.25 oikealla). Ensimmäistä tapaa pidetään yleisesti parempana, vaikka se lisääkin hyppyjen määrää palvelimen ja tallennuslaitteen välillä. Runkokytkimeen kiinnitetty tallennuslaite kuitenkin varaa portteja, joihin voitaisiin kiinnittää myös uusi reunakytkin. Toinen tapa rajoittaa tallennusverkon skaalautuvuutta. Mikäli runkokytkiminä käytetään director-luokan kytkimiä, joissa porttien määrä on suuri, tämä ei välttämättä muodostu ongelmaksi. [5, s. 156-158]

2.7 Tallennusverkon hallinta ja monitorointi

Perinteisen verkonhallinnan tärkein tavoite on taata turvallinen datan kuljetus verkkoinfrastruktuurin läpi. Se mitä datalle tapahtuu sen saavuttua kohteensa, ei yleensä koske enää verkonhallintaa. Tallennusverkon hallinta laajentaa perinteisen verkon hallinnan tavoitteita. Tallennusverkossa sekä datan turvallinen kuljetus että säilöntä ovat päätavoitteita. [2, s. 223]

2.7.1 Keskitetty hallinta

Tallennusverkot ovat erittäin monimutkaisia kokonaisuuksia, jotka koostuvat sadoista tai jopa tuhansista erillisistä laitteista. Useimmilla näistä laitteista on oma hallintakonsoli, jonka avulla laitetta voi konfiguroida ja diagnosoida. Kokonaisuuden hallinta muodostuu erittäin nopeasti vaikeaksi, kun tallennusverkon koko kasvaa. Useiden erillisten hallintakonsolien käyttö päivittäisessä työssä monimutkaistaa verkon hallintaa ja kasvattaa ylläpitoon käytettäviä kustannuksia. Tällaisen kokonaisuuden hallinta ja monitorointi vaatii suunnitelmallisuutta sekä keskitettyä hallintajärjestelmää, jolla pystytään hahmottamaan kokonaisuuksia ja josta jokaista yksittäistä laitetta pystytään hallinnoimaan. Keskitetyn hallintakonsolin tulee pystyä kommunikoimaan kaikkien tallennusverkon laitteiden kanssa, jotta koko verkon hallinta ja monitorointi voidaan suorittaa samasta paikasta. [1, s. 388-389; 3, s. 283]

Jotta keskitetty hallintajärjestelmä olisi mahdollisimman tehokas, sen on kyettävä kommunikoimaan mahdollisimman monien eri laitevalmistajien laitteiden kanssa. [3, s. 284] Keskitetystä hallintajärjestelmästä on lisäksi hyvä löytyä seuraavat viisi peruspalvelua, joilla tallennusverkon hallintaa voidaan helpottaa:

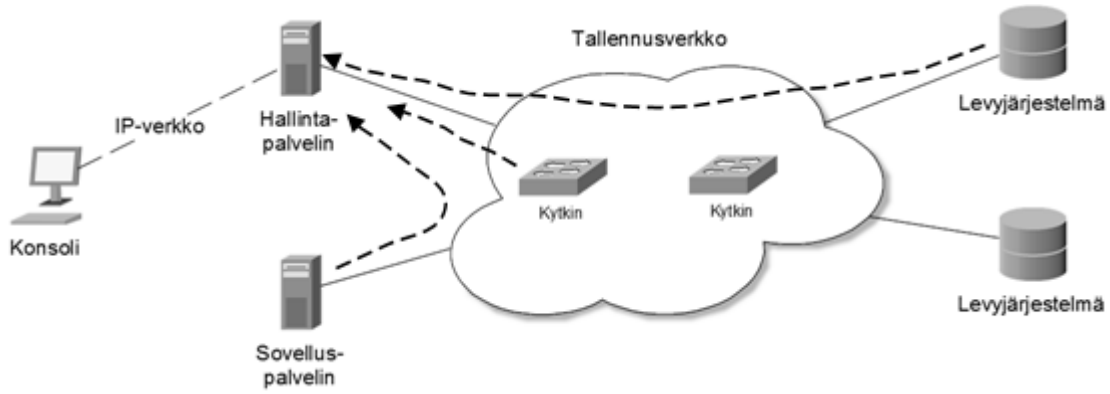
1. Verkon laitteiden automaattisella löytämisellä (discovery) voidaan kerätä laitteiden tiedot, asetukset ja tilatieto. Näiden tietojen avulla voidaan muodostaa verkon topologia.
2. Monitorointijärjestelmä valvoo verkon sovellusten ja resurssien tilaa sekä tekee automaattisia ilmoituksia mahdollisista vioista.

3. Keskitetyllä hallintajärjestelmällä pitää pystyä konfiguroimaan tallennusverkon laitteita samasta paikasta. Verkon konfigurointi nopeutuu huomattavasti, kun jokaista laitetta ei pidä konfiguroida erillisestä konsolista.
4. Keskitetty hallintajärjestelmä kerää tilastotietoa suorituskyvystä, virheistä ja konfiguraatiosta ja suorittaa niiden analysointia. Näin pystytään analysoimaan verkon toimintaa pitkällä aikavälillä.
5. Tallennusverkon hallinnalle ominainen keskitetty datan hallinta valvoo kaikkea tiedon tallentamiseen ja saatavuuteen liittyviä sovelluksia ja resursseja.

Hallintakonsoli kommunikoi tallennusverkon komponenttien kanssa erilaisten rajapintojen ja protokollien kautta. Nämä voidaan jakaa kanavansisäisiin ja ulkopuolisiin hallintamekanismeihin.

2.7.2 Kanavansisäinen hallinta

Tallennusverkoissa kanavansisäisellä hallinnalla (in-band management) tarkoitetaan verkon hallintaa, joka suoritetaan samaa kanavaa pitkin, jota itse datakin kuljetetaan. Datan kuljettamisen lisäksi laitteen löytäminen, monitorointi ja konfigurointi voidaan suorittaa samaa kanavaa pitkin (kuva 2.26).



Kuva 2.26. Kanavansisäinen hallinta kerää hallintadatan tallennusverkon kautta

Kanavansisäiset palvelut voidaan jakaa toiminnallisiin ja hallinnallisiin palveluihin. Toiminnallisilla palveluilla tarkoitetaan datan siirtämiseen tarkoitettuja toimintoja ja hallinnallisilla palveluilla laitteiden monitorointia ja konfigurointia. Molempia voidaan käyttää hallintatarkoituksiin. Fibre Channelin standardit FC-MI ja FC-GS muodostavat pohjan kanavansisäiselle hallinnalle Fibre Channelia käyttävälle tallennusverkolle. FC-MI-standardi määrittelee perusoperaatiot, jotka takaavat eri laitteiden välisen yhteensopivuuden sekä perusvaatimukset, jotka laitteen on täytettävä, että se voi käyttää kanavansisäistä hallintaa. FC-GS-standardi määrittelee Fibre Channelin hallintapalvelut. Hallinnan ja monitoroinnin kannalta tärkeimmät kudoksen palveluista ovat nimi-, konfiguraatio- ja vyöhykepalvelin. Nimipalvelin on osa toiminnallisia palveluita ja sen avulla saadaan tietoa yhteyksien välisistä riippuvaisuuksista sekä porttien asetukset. Konfiguraatiopalvelin kuuluu hallinnallisiin palveluihin ja sen avulla saadaan selville

tallennusverkon topologia. Vyöhykepalvelin kuuluu sekä hallinnallisiin että toiminnallisiin palveluihin, koska sen avulla voidaan konfiguroida ja selvittää vyöhykejakoja. [1, s. 396-397]

Kanavansisäinen hallinta yksinkertaistaa tallennusverkon käyttöön siirtymistä, koska erillistä verkkoa laitteiden hallintaan ei tarvita. Kaikilla tallennusverkon laitteilla on automaattisesti liityntä, jolla laitetta voidaan hallita. Suurin ongelma kanavansisäisessä hallinnassa on hallinnan menettäminen, mikäli jokin osa tallennusverkossa hajoaa. Tällöin kyseistä laitetta ei pystytä monitoroimaan tai diagnosoimaan, koska data ja hallintatieto kulkevat samaa kanavaa pitkin. Kahdennetuilla linkeillä voidaan kiertää tätä ongelmaa, mutta sekään ei poista ongelmaa kokonaan ja se lisää myös tallennusverkosta aiheutuvia kustannuksia. Kanavansisäinen hallinta ei pysty tehokkaasti valvovaan tallennusverkon fyysisellä tasolla tapahtuvia virheitä. [2, s. 225-226]

2.7.3 Kanavan ulkopuolinen hallinta

Kanavan ulkopuolisessa hallinnassa hallintaan käytetään tallennusverkon liitynnän sijasta jotakin toista liityntää, esimerkiksi lähiverkkoa. Tällä pystytään välttämään kanavansisäisen hallinnan ongelma, jossa hallintayhteys menetetään laitevian tai linkin katkeamisen takia. Haittapuolena kanavan ulkopuoleisessa hallinnassa on se, että sen avulla ei pystytä suoraan määrittelemään verkon topologiaa. Siihen tarvitaan kanavansisäistä hallintaa. [2, s. 226-227] Turvallisuussyistä hallintaan voidaan joutua myös käyttämään omaa lähiverkkoyhteyttä, mikä lisää tallennusverkon kokonaiskustannuksia. [1, s. 398]

Kanavanulkopuoleisessa hallinnassa käytetyimmät protokollat ovat SNMP, WBEM/CIM ja SMI-S. Tämän lisäksi voidaan käyttää SSH:ta tai laitespesifisiä hallintaliityntöjä. Standardointiorganisaatiot kuten SNIA, IETF ja FCIA työstävät standardia hallintaliitynnöistä. Tämän tavoitteena on saada aikaiseksi valmistajasta riippumattomia hallintajärjestelmiä. [1, s. 393-394]

SNMP

SNMP on ollut pitkään käytetyin protokolla verkkojen hallintaan. Sen ensimmäinen versio julkaistiin jo vuonna 1988. SNMP ei ole menettänyt suosiotaan verkonhallintaprotokollana, vaikka uudempiakin vaihtoehtoja on ollut jo pitkään. Tämä johtuu pitkälti SNMP:n yksinkertaisesta arkkitehtuurista. [1, s. 399]

SNMP-hallintasovellusta kutsutaan NMS:ksi (Network Management System) [1, s.400] ja sen tehtävä on kerätä SNMP-viestejä muilta verkon laitteilta. Laitteilla, joita NMS valvoo, täytyy olla asennettuna SNMP-agentti, joka välittää viestejä hallittavalta järjestelmältä NMS:lle ja toisin päin. Hallittava tieto on organisoituna hierarkiseen tietorakenteeseen jota kutsutaan MIB:ksi (Management Information Base). Tavoite on, että valmistajat suunnittelevat laitteensa niin, että ne ovat yhteensopivia olemassa

olevien MIB:ien kanssa. Näin eri valmistajien välinen hallinta on mahdollista. [10, s. 136]

SNMP perustuu IP-protokollan käyttöön ja määrittelee neljä erilaista komentoa, joilla voidaan hakea ja lähettää tietoa NMS:n ja hallittavien laitteiden välillä. Get-viestillä NMS voi pyytää agentilta yhtä tai useaa asetusta. GetNext-viestillä NMS voi pyytää suoraan seuraavaa arvoa Get-viestin jälkeen. Set-viestillä NMS voi asettaa uusia asetuksia hallittavaan kohteeseen. Trap-viestillä agentti voi ilmoittaa NMS:lle järjestelmässä tapahtuneista muutoksista. [1, s. 401; 10, s. 137]

Tallennusverkkojen hallintaan on kehitetty kaksi MIB:iä. Ensimmäinen niistä on Fibre Channel Fabric Element MIB, joka on tarkoitettu erityisesti tallennusverkon kytkimille. Sen avulla voidaan noutaa tietoja kytkimen porteista ja niiden suorituskyvystä. Toinen on Fibre Channel Management MIB, jonka avulla saadaan tietoa linkeistä ja vyöhykkeistä. Näin pystytään päättelemään verkon fyysistä ja loogista rakennetta. Sen avulla päästään myös käsiksi kudoksen nimipalvelimeen, josta verkon topologia saadaan myös mallinnettua. [1, s. 401; 10, s. 138]

WBEM ja CIM

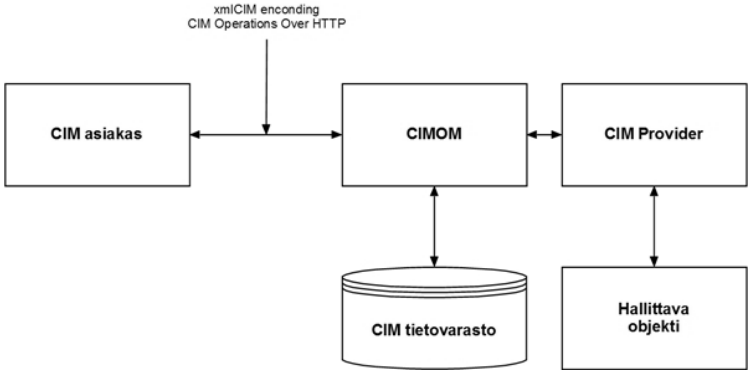
WBEM on DMTF:n kehittämä sovelluskehys, jonka tarkoitus on yhdistää kaiken tyyppisten järjestelmien (tallennuslaitteiden, palvelimien, kytkimien, siltojen, palomuurien ja sovellusten) hallinta saman protokollan alle [4, s. 398]. WBEM käyttää tunnettuja web-tekniikoita kuten XML ja HTTP, datan kuljetukseen ja esittämiseen hallittavien järjestelmien ja hallintasovelluksen välillä. Tämän lisäksi WBEM:stä löytyy rajapinnat, joiden avulla voidaan käyttää perinteisiä hallintaprotokollia kuten SNMP. [1, s. 404]

WBEM määrittelee kolme osaa, jotka standardoivat liitynnät hallittavien järjestelmien ja hallintatyökalujen välillä. WBEM käyttää CIM:iä mallintamaan objekteja, samalla tavalla kuin SNMP käyttä MIBiä. CIM perustuu olioiden mallintamiseen (Object Orientated Modelling). OOM:ia käytetään mm. olio-ohjelmoinnissa, mutta se ei ole ohjelmointikieli vaan kieli, jolla voi mallintaa olioita. CIM perustuu siis luokkajakoon, jossa aliluokat perivät kantaluokaltaan attribuutit ja metodit. [1, s. 405] Laitevalmistajat voivat tehdä omia aliluokkiaan jo olemassa oleville luokille vaarantamatta valmistajien välistä yhteensopivuutta [4, s. 398].

WBEM on toteutettu modulaarisesti, jotta CIM:n mallintamiseen tai datan kuljettamiseen voidaan käyttää mitä tahansa tekniikkaa. Yleisin käytetty tekniikka CIM:n mallintamisessa on XML, joka on määritelty *xmlCIM*-standardissa. [4, s.398] Se määrittelee XML-formaatit, joissa CIM-objektien metodeja kutsutaan ja vastaavasti, miten niihin vastataan. Tämän avulla kaksi WBEM-sovellusta voi keskustella riippumatta siitä, miten ne on toteutettu alemmilla tasoilla. [1, s. 405]

HTTP on käytetyin tiedonsiirtoprotokolla WBEM:n käytössä [4, s. 398]. Sen toiminta on määritelty *CIM Operations Over HTTP*-spesifikaatiossa. Se määrittelee, miten WBEM-sovellusten lähettävät xmlCIM-viestejä toisilleen, joiden avullla CIM-objektit voivat kutsua toistensa metodeja. [1, s. 405]

Edelliset standardit määrittelevät, miten ja millaisessa muodossa data siirretään WBEM-sovellusten välillä. Näiden lisäksi keskeisiä WBEM-arkkitehtuurin osia ovat CIM Object Manager (CIMOM) ja CIM Provider (kuva 2.27). CIM Provider toimii CIMOM:n ja hallittavan objektin välillä. Sen tehtävä on välittää hallintadataa hallittavalta objektilta CIMOM:lle. CIM Provider sisältää rajapinnat muihin hallintaprotokolliin. Se voi käyttää laitteen hallintaan esimerkiksi SNMP:tä ja esittää SNMP:llä saamansa tiedot CIM-objekteina. CIMOM toimii välittäjänä CIM Providerin ja hallintasovelluksen välillä (CIM client). CIMOM sisältää tietovaraston CIM-objektien malleista.



Kuva 2.27. WBEM/CIM-arkkitehtuuri

SMI-S

WBEM/CIM-arkkitehtuuri on osoittautunut hyväksi homogeenisten verkkojen hallintaan. Se ei kuitenkaan yksistään ole riittävä tallennusverkkojen hallintaan. SMI-S on SNIA:n kehittämä laajennus olemassa olevaan WBEM/CIM-arkkitehtuuriin, jonka tarkoitus on parantaa sen sopivuutta tallennusverkkojen hallintaan. SMI-S:n tavoite on kehittää avoin, valmistajasta riippumaton sovellusrajapinta (API), jonka avulla voidaan suorittaa tallennusverkon hallinnan kaikki toimenpiteet. [1, s. 409]

SMI-S laajentaa WBEM-arkkitehtuuria kahdella tavalla. Ensinnäkin se laajentaa CIM-objektit tallennusverkon hallintaan sopiviksi. Toiseksi se lisää WBEM-arkkitehtuuriin kaksi uutta palvelua, jotka ovat Directory Manager ja Lock Manager. Directory Managerin tarkoitus on yksinkertaistaa tallennusverkon laitteiden löytämistä (discovery). Lock Managerin avulla voidaan CIM-objekteja suojata monen hallintasovelluksen samanaikaiselta käytöltä tai asettaa CIM-objekteille eritasoisia suojauksia. Hallintasovellusten on tarkistettava oikeutensa Lock Managerilta ennen kuin ne voivat käyttää suojattuja CIM-objekteja. [1, s. 410] Lock Managerin määrittely on vielä kesken [12].

SSH

SSH (Secure Shell) on protokolla, jonka avulla kaksi verkkolaitetta voi vaihtaa tietoa turvallisen tunnelin läpi turvattomassa verkossa. Se suunniteltiin korvaamaan telnet, rlogin ja rsh, jotka ovat selkokielisiä protokollia. Näitä protokollia käytetään pääasiassa laitteiden komentotulkin etäkäyttöön. Käyttäjätunnukset, salasanat ja komennot kulkevat suojaamattomana ja selkokielisenä esimerkiksi Internetin yli, mikä altistaa ne tiedon urkinnalle. SSH salaa liikenteen verkkolaitteiden välillä muodostaen suojatun tunnelin laitteiden välille. Palvelimen ja käyttäjän autentikointiin SSH käyttää julkisen avaimen kryptografiaa. [13]

SSH:n on alun perin kehittänyt suomalainen Tatu Ylönen. Hän kehitti protokollan sen jälkeen kun Teknillinen korkeakoulu joutui salasanojen urkintahyökkäyksen kohteeksi. SSH:n ensimmäinen versio julkaistiin vuonna 1995 ja sillä oli vuoden loppuun mennessä jo 20000 käyttäjää. Kasvavan suosion takia Ylönen kaupallisti SSH:n ja perusti SSH Communications Ltd:n. Kaupallistumisen myötä SSH:sta tuli tiukasti lisensoitu tuote, joka johti avoimen OpenSSH:n kehittämiseen. OpenSSH on tällä hetkellä maailman käytetyin SSH-palvelinsovellus. [14] Tatu Ylösen kehittämää SSH:ta kutsutaan nykyään SSH-1:ksi. Nykyään käytössä oleva SSH-2 on IETF:n määrittelemä standardi, joka parantaa SSH:n turvallisuutta ja suorituskykyä. [13]

SSH-2:n arkkitehtuuri koostuu kuljetuskerroksesta (transport layer, RFC4253), käyttäjän autentikointikerroksesta (user authentication layer, RFC4252) ja yhteyskerroksesta (communication layer, RFC4254). Kuljetuskerros vastaa avaimien vaihdosta verkkolaitteiden välillä ja muodostaa salatun ja pakatun yhteyden niiden välille. Autentikointikerros varmistaa, että laitteet voivat luottaa toisiinsa yhteyden muodostamiseksi. Autentikointiin voidaan käyttää salasanaa, julkiseen avaimeen perustuvaa autentikointia tai kertakäyttösalasanoja, kuten S/Key tai SecurID. Yhteyskerros muodostaa SSH:tä käyttävien laitteiden välillä loogisia kanavia. Yhdessä SSH-yhteydessä voidaan kuljettaa useaa loogista kanavaa samanaikaisesti, joista jokainen voi lähettää ja vastaanottaa dataa. [13]

Komentojen suorittamisen lisäksi SSH:ta voidaan käyttää tiedostojen siirtämiseen. SSH:hon perustuvia tiedostonsiirtoprotokollia ovat mm. SFTP ja SCP. SFTP perustuu löyhästi FTP:n käyttöön SSH:n yli ja se sisältyy SSH-2 standardiin. SCP (Secure Copy) vastaavasti perustuu vanhempaan salaamattomaan RCP-protokollaan (Remote Copy). Molemmat ovat yleisesti käytettyjä tiedostonsiirtoprotokollia. [14]

Vaikka SSH ei olekaan varsinaisesti laitteiden hallintaan tai monitorointiin tarkoitettu protokolla, se mahdollistaa monipuolisen laitteiden hallinnan. SSH:n avulla voidaan kerätä tietoa suoraan käyttöjärjestelmästä ajamalla komentorivikäyttöliittymän komentoja SSH:n läpi. Sillä voidaan kopioida tiedostaja hallittaville laitteille ja ajaa esimerkiksi ohjelmistopäivityksiä. SSH:n mahdollistaa itse tehtyjen räätälöityjen

hallintasovellusten ajamisen hallittavissa laitteissa, jolla voidaan automatisoida toimenpiteitä.

2.8 Muut tavat toteuttaa tallennusverkko

Fibre Channel on tällä hetkellä suosituin tapa toteuttaa tallennusverkko, koska se soveltuu ominaisuuksiltaan erittäin hyvin tähän käyttötarkoitukseen. Sen ongelma on kuitenkin eri laitevalmistajien välinen yhteensopivuus, joka rajoittaa asiakkaiden valinnanvapautta ostopäätöksiä tehdessä. Fibre Channelia käyttävät verkkotuotteet ovat myös huomattavasti kalliimpia verrattuna esimerkiksi lähiverkon kytkimiin ja IP-reitittimiin. Fibre Channelin lisäksi tallennusverkko voidaan rakentaa erilaisilla IP-protokollaan ja Ethernet-tekniikkaan perustuvilla ratkaisuilla. Täysin erilainen lähestymistapa tallennusverkon rakentamiseen on InfiniBand.

2.8.1 IP-pohjaiset järjestelmät

IP-pohjaisten tallennusverkkojen etuna on se, että sekä tallennusverkon että lähiverkon data voidaan kuljettaa saman kuljetusprotokollan avulla. Näiden kuljettamiseen voidaan tarvittaessa käyttää samaa verkkoliityntää ja -infrastruktuuria, jolla voidaan vähentää tallennusverkosta aiheutuvia kustannuksia. Tärkeimmät IP-pohjaiset tallennusverkkoprotokollat ovat FCIP, iFCP ja iSCSI.

FCIP

FCIP (Fibre Channel over TCP/IP) ei ole suunniteltu korvaamaan Fibre Channelia, vaan sitä voidaan pitää laajennuksena Fibre Channelille. FCIP on määritelty RFC3821:ssa [15] ja se kuvaa mekanismit, jolla Fibre Channel -saarekkeita voidaan yhdistää IP-pohjaisen verkon ylitse muodostaen yhden Fibre Channel-kudoksen. FCIP tukeutuu IP-pohjaisiin verkkopalveluihin, jotka takaavat yhteyden Fibre Channel -saarekkeiden välillä joko LAN, MAN tai WAN-verkon ylitse. FCIP:n tarkoitus on kuljettaa Fibre Channel-liikenne IP-verkon yli siten, että Fibre Channel-laitteet kudoksissa ovat täysin tietämättömiä IP-verkosta. [15] FCIP kapseloi Fibre Channelin kehykset ja kuljettaa ne TCP/IP-tunnelia pitkin. Vuonvalvonta ja virheidenhallinta jätetään TCP:n huoleksi. [11]

Vaikka Fibre Channel pystyy erittäin pitkiin siirtoetäisyyksiin, ei se silti pysty vastaamaan Internetin tiedonsiirtomahdollisuuksiin. FCIP onkin erittäin käytetty esimerkiksi varmuuskopioinnissa. Levyjärjestelmät tai kokonaiset datakeskukset voidaan kahdentaa useiden satojen kilometrien päähän. FCIP:n suosiota lisää myös se, että se ei vaadi muutoksia olemassa oleviin tallennusverkkoihin, koska Fibre Channel kehykset viedään muuttumattomana IP-tunnelin yli.

iFCP

iFCP (Internet Fibre Channel Protocol) on yhdyskäytäväprotokolla (gateway-to-gateway protocol), jonka avulla voidaan muodostaa Fibre Channel -laitteille

tallennusverkko käyttämällä TCP/IP-verkkoa. Sen avulla voidaan siis tarjota Fibre Channel-laitteille tallennusverkko ilman Fibre Channel-kudosta ja kudoksen hallintaa. [2, s. 157] iFCP on määritelty RFC4172:ssa [16]. Verkkoon kytketyillä Fibre Channel-laitteille iFCP:llä muodostettu kudos on täysin läpinäkyvä. iFCP-laitteiden pitää pystyä emuloimaan kudoksen palveluja (nimipalvelin, vyöhykkeet), jotta tämä läpinäkyvyys saavutetaan. [16]

iFCP-yhdyskäytävän portit toimivat kahdessa eri roolissa. Fibre Channel N_Portti kytkeytyy iFCP-yhdyskäytävässä normaalisti kudoksen F_Porttiin. Yhdyskäytävän pitää siis tukea kudokseen sisäänkirjautumista ja nimipalvelua, jotta N_Portilla kytkeytynyt laite luulee olevansa normaalissa kudoksessa. iFCP-yhdyskäytävän portti pitää toimia myös IP-protokollaa tukevassa roolissa. Sen pitää osata muuttaa Fibre Channelin liikenne IP:ksi ja toisin päin. Portti muuttaa 24-bittisen kudososoitteen uniikiksi 32-bittiseksi IP-osoitteeksi, jolla laitteet tunnistetaan IP-kudoksessa. Muunnoksen jälkeen iFCP-paketti voidaan reitittää normaalisti IP-verkossa. [2, s. 157] iFCP käyttää TCP:n ruuhkanhallintaa ja virheidenhallintamekanismeja takaamaan luotettavan yhteyden verkon laitteiden välillä. [16]

iFCP:n etu verrattuna tavalliseen Fibre Channeliin ja FCIP:n käyttöön on se, että iFCP:llä voidaan muodostaa autonomisia alueita, jotka tarjoavat yhteyden tallennusverkon laitteiden välillä, mutta silti eristävät alueet omiksi kokonaisuuksikseen. iFCP ei välitä Fibre Channel -kytkimien tapaan tietoa keskenään alueelta toiselle, joten alueet eivät sulaudu yhdeksi kudokseksi. iFCP ei myöskään muodosta tunneleita yhdyskäytävien välille, vaan runkoverkossa voidaan käyttää normaalia IP-reititystä. [2, s. 158]

iFCP ei ole kuitenkaan yleistynyt teollisuudessa monimutkaisuutensa vuoksi. FCIP on huomattavasti helpompi tapa ulottaa tallennusverkko eri datakeskusten välille. [2, s. 171]

iSCSI

iSCSI:n perusidea on kuljettaa SCSI-komentoja suoraan TCP/IP-verkon ylitse. iSCSI:n perustoiminta on määritelty RFC3720:ssä [17]. iSCSI:n lähestymistapa on siis samankaltainen kuin Fibre Channelissa, mutta iSCSI käyttää TCP/IP- ja Ethernet-tekniikkaa komentojen välittämiseen. iFCP tarjoaa migraatiota Fibre Chanellin ja IP-verkon välillä, mutta iSCSI:ssa päätelaitteet ovat puhtaasti iSCSI-laitteita, joiden välillä on IP-reititysverkko.

iSCSI:n suurin eroavaisuus normaaliin tallennusverkoon on se, että iSCSI-laitteiden pitää olla paljon älykkäämpiä verrattuna esimerkiksi Fibre Channel -laitteisiin. Fibre Channelissä verkko on älykäs ja siihen liittyvät laitteet enemmän tai vähemmän passiivisia. iSCSI:ssa voidaan käyttää normaaleja IP-reittittimiä ja Ethernet-kytkimiä,

joilta ei voida olettaa tallennusverkolta vaadittavaa älykkyyttä. Siksi älykkyys on iSCSI:ssa siirretty päätelaitteille.

iSCSI-laitteilla täytyy olla yksiselitteinen IP-osoite, jolla ne pystyvät tunnistamaan toisensa verkossa. IP-osoite voidaan määritellä staattisesti tai se voidaan jakaa DCHP:n avulla. iSCSI-laitteiden täytyy myös suorittaa sisäänkirjautuminen kuten Fibre Channelissa. iSCSI:ssa sisäänkirjautuminen suoritetaan kuitenkin suoraan toisen iSCSI-laitteen kanssa, eikä verkon kanssa. [2, s. 160-163]

iSCSI mahdollistaa puhtaasti IP-protokollaan perustuvan tallennusverkon rakentamisen. iFCP ja FCIP nojautuvat edelleen Fibre Channel-protokollaan. iSCSI:n avulla datakeskuksessa ei tarvita kahta erillistä verkkoa ”normaalin” datan ja tallennusverkon datan kuljettamiseen. Tämä lähestymistapa mahdollistaa myös IP-pohjaisen verkon reitityksen ja hallintatyökalujen käytön. Näin tallennusverkon rakentamisesta aiheutuvat kustannukset saadaan pudotettua huomattavasti pienemmiksi verrattuna esimerkiksi Fibre Channel-tallennusverkkoon. iSCSI:lla ei myöskään ole minkäänlaista fyysisen linkin pituusrajoitetta. Saman verkon käyttö tosin lisää suorituskykyongelmia ja heikentää tietoturvaa. [10, s. 93]

2.8.2 FCoE

Vaikka Fibre Channel on tällä hetkellä ylivoimaisesti käytetyin teknologia tallennusverkoissa, se tuskin tulee olemaan aina näin. Yksi kovimmista haastajista Fibre Channelille on kehitteillä oleva FCoE-teknologia (Fibre Channel over Ethernet). FCoE kapseloi Fibre Channel-kehykset Ethernet-kehyksen sisälle. Se on ANSI T11- komitean kehittämä runkoverkko-teknologia tallennusverkkoihin ja se kuuluu osana FC-BB-5-projektiin [18]. FCoE:n kehittämiseen on kaksi painavaa syytä. Vaikka Fibre Channel-tuotteet ovat halventuneet viimevuosina, ovat ne edelleen silti kalliita. Toinen syy on 10 gigabitin Ethernet-standardin valmistuminen ja sen odotetaan syrjäyttävän yhden Gigabitin Ethernetin lähivuosina [1, s. 125].

Kuten kaikki uusi teknologia yleensä, myös FCoE tähtää kustannusten pienentämiseen ja tehokkuuden kasvattamiseen. FCoE:n avulla datakeskuksessa voidaan luopua usean eri verkkoinfrastruktuurin ylläpidosta, koska sekä ”tavallinen” verkkoliikenne että tallennusverkon liikenne voidaan siirtää samaan verkkoon. FCoE:tä ei kuitenkaan voida suoraan implementoida Ethernet-verkkoon, vaan ensin on ratkaistava Ethernetin ongelmat koskien tallennusverkon liikennettä. Tällaisesta kehittyneestä Ethernet-verkosta käytetään nimitystä FCoCEE (Fibre Channel over Convergence Enhanced Ethernet). [19, s. 2]

FCoE:tä käytettäessä tulee ensimmäiseksi vaihtaa verkkosovittimet. Uusien verkkoasovittimien pitää toimia sekä Fibre Channelin että normaalin verkkoliikenteen välittäjinä. Palvelimissa tällaisesta adapterista käytetään nimitystä CNA (Converged Network Adapter). CNA näkyy käyttöjärjestelmälle sekä Fibre Channel HBA:na sekä

normaalina verkkosovittimena, jonka kautta lähiverkko sekä klusterointiliikkenne kulkee. [19, s. 4] Käyttöjärjestelmälle muutos on täysin läpinäkyvä, koska FCoE korvaa ainoastaan Fibre Channelin kerrokset FC-0 ja FC-1 [19, s. 6].

FCoCEE:ssä normaaliin Ethernet-tekniikkaan on lisättävä vuonvalvontamekanismi, jotta kehyksiä ei aleta tiputtaa verkon ruuhkautuessa. Tämä pystytään toteuttamaan PAUSE-mekanismilla joka on määritelty IEEE 802.3 Annex 31B:ssä. Ethernet-verkkolaite, jonka puskuri on täynnä, alkaa normaalisti pudottaa paketteja. PAUSE-mekanismilla verkkolaite voi ilmoittaa lähettäjille, että niiden on lopetettava pakettien lähettäminen siihen asti, kunnes verkkolaite antaa niille luvan. PAUSE-mekanismin ongelma on se, että silloin se katkaisee kaiken liikenteen. IEEE työstää parannusta tähän ja tulevaisuudessa tullaan mahdollisesti näkemään prioriteettiin perutuva vuonvalvontamekanismi Ethernet-verkkossa. [19, s. 6]

FCoE poistaa fyysiset Fibre Channel -portit, joten se aiheuttaa muutoksia Fibre Channel-termistöön. Porteista tulee virtuaalisia, joten erityyppisten porttien nimiin lisätään V-kirjain erottamaan ne fyysistä vastineistaan (VN_Port, VF_Port, VE_Port jne). Myös Fibre Channel-linkeistä tulee virtuaalisia, koska FCoE mahdollistaa yhteyden yhdestä VN_Portista useaan VF_Porttiin. Yhteen VF_Porttiin voi myös liittyä useita VN_Portteja. FCoE:tä tukevasta päätelaitteesta käytetään myös nimeä ENode. Yhdessä ENodessa voi olla yksi tai useampi CNA, jossa voi olla yksi tai useampi FCoE_LEP (FCoE Link End Point) ja virtuaalinen portti. [19, s. 9-10]

FCoE:n avulla datakeskuksen kuluja voidaan karsia samalla säilyttäen silti Fibre Channelista tutut palvelut. Laajoissa datakeskuksissa usean eri verkon ylläpitäminen aiheuttaa suuria kustannuksia, joten siirtyminen ainoastaan yhden verkon käyttöön tuntuu järkevältä. FCoE on kuitenkin vasta tulossa markkinoille, joten sen käytettävyydestä ei ole vielä paljon kokemusta. FCoE:n aloittama trendi verkkojen yhdistämisestä on varmasti suunta, mihin kehitystä ollaan viemässä.

2.8.3 Infiniband

Viimevuosien aikana on tullut selväksi, että nykyinen jaettuun väylään perustuva arkkitehtuuri muodostuu palvelimien pullonkaulaksi. InfiniBand pyrkii muuttamaan tätä lähestymistapaa ja se on täysin uusi lähestymistapa tallennusverkon muodostamiseen. [20] InfiniBand korvaa palvelimen PCI-väylän sarjamuotoisella kytketyllä kudoksella. InfiniBandissa tallennusverkko ulottuu siis syvälle päätelaitteden sisälle asti. [1, s. 118]

InfiniBandissa kaikki yhteydet päätelaittaitteiden, kytkimien ja reitittimien välillä ovat sarjamuotoisia pisteestä-pisteeseen-yhteyksiä. Sarjamuotoinen yhteys on parempi kuin PCI:n rinnakkaisväylä. Pisteestä-pisteeseen-yhteydet takaavat täyden kapasiteetin kahden päätelaitteen välillä, koska linkki niiden välillä varataan niiden käyttöön. Linkin varauksella vältetään kilpailu väylästä sekä viiveet, joita aiheutuu jaetun väylän käytössä kovassa kuormassa. [20]

InfiniBandin yksi fyysinen linkki pystyy siirtämään dataa 2,5 Gbit/s. Linkkejä voidaan myös niputtaa neljän ja kahdentoista linkin nipuiksi, millä saavutetaan 10 ja 30 Gbit/s siirtonopeudet. Fyysisiin yhteyksiin voidaan käyttää joko kuparisia tai optisia kaapeleita. Kuparisten kaapeleiden maksimipituus on 17 metriä ja optisten 10 kilometriä. [1, s. 119]

Päätelaitteiden sovittimia kutsutaan InfiniBandissa kanavasovittimiksi (Channel Adapter). Sovittimet jaetaan kahteen kategoriaan. Palvelimen kanavasovitinta kutsutaan HCA:ksi (Host Channel Adapter), ja sen tehtävä on yhdistää käyttöjärjestelmä InfiniBand-verkkoon. Levyjärjestelmien ja muiden tallennuslaitteiden sovittimia kutsutaan TCA:ksi (Target Channel Adapter). InfiniBand tukee myös useita reittejä palvelimien ja tallennuslaitteiden välillä, jolla saadaan aikaan kahdennettuja yhteyksiä. [20]

InfiniBandissa palvelimet supistuvat prosessoreja ja muistia sisältäviksi moduuleiksi, jotka prosessoivat dataa, jota tallennetaan ja haetaan InfiniBandin avulla levyjärjestelmiltä. Klustereiden muodostaminen InfiniBandin avulla on helppoa. Klusteriin kuuluvat palvelimet voidaan yhdistää suoralla linkillä, jota kautta klusterin tilatietoa voidaan välittää.

Tallennuslaitteet tulevat koko ajan nopeammiksi. InfiniBand mahdollistaa nopean kommunikaation palvelimien ja tallennuslaitteiden välillä vähentäen prosessorille aiheutuvaa kuormaa. [1, s. 119] InfiniBand on saavuttanut valtavasti tukea teollisuudelta, mikä on mahdollistanut sen nopean tulon markkinoille heti sen spesifikaation valmistuttua [20].

3 KEHITETTÄVÄ YMPÄRISTÖ

Tämän tutkimuksen kohteena oleva datakeskus sisältää yhden Suomen suurimmista tallennusverkoista. Kuituporttien määrä tallennusverkossa on noin 2000 kappaletta. Datakeskus toimii palvelinohjelmiston kehitysympäristönä. Ohjelmistoa ajetaan kahdesta tai useammasta palvelimesta koostuvassa klusterissa. Palvelimien käyttöjärjestelmänä toimii Red Hat Enterprise Linux (RHEL). Klusterien yhteinen tallennustila on sijoitettu tallennusverkon levyjärjestelmille, jotta klusterin kaikilla palvelimilla on pääsy niihin.

Tallennusverkon koko on kasvanut viime vuosina erittäin paljon, minkä seurauksena tarve keskitetylle hallinta- ja monitorointijärjestelmälle on syntynyt. Hallintaohjelmistoksi tallennusverkkoon on valittu Hewlett-Packardin valmistama HP Storage Essentials SRM Enterprise Edition, koska se on jatkuvasti kehittyvä, suosittu ja kansainvälisesti arvostettu ohjelmisto tallennusverkon hallintaan. Tallennusverkko koostuu myös suurilta osin HP:n valmistamista palvelimista ja levyjärjestelmistä, joten tämäkin on puoltanut kyseisen ohjelmiston valintaa.

3.1 Hallinnassa käytettävät ohjelmistot

HP Storage Essentials (SE) ei pysty yksin keräämään kaikkea tietoa tallennusverkosta tai hallitsemaan sen laitteita suoraan. Jotkin laitteet, kuten kytkimet ja eräät levyjärjestelmät, vaativat ohjelmiston, joka välittää tietoa SE:n ja laitteen välillä. Brocaden valmistamat tallennusverkon kytkimet tarvitsevat Brocade SMI Agent-ohjelmiston, jonka kautta kytkimiä koskeva tieto välitetään SE:lle. HP EVA -levyjärjestelmien (Enterprise Virtual Array) hallintaan voidaan käyttää HP Command View EVA-ohjelmistoa. Sen kautta HP EVA-levyjärjestelmien hallinta voidaan integroida SE:iin. Tämän lisäksi tallennusverkon hallintaan käytetään HP Systems Insight Manager- sekä HP SRM Report Optimizer -ohjelmistoja.

3.1.1 HP Storage Essentials SRM Enterprise Edition

HP Storage Essentials SRM (SE) on keskitetty hallintasovellus tallennusverkkojen hallintaan. Sen avulla voidaan hallita heterogeenisiä tallennusverkkoja, jotka koostuvat useiden valmistajien valmistamista laitteista. SE:llä voidaan hallita myös DAS- ja NAS-laitteita [21]. Storage Essentials käyttää avoimia standardeja, kuten WBEM:iin ja CIM:ään perustuvaa SMI-S:ää. [22, s. 1]

SE koostuu kolmesta erillisestä sovelluksesta, jotka ovat hallintapalvelin, tietokantasovellus sekä CIM laajennukset. Hallintapalvelin on sovellus, joka tarjoaa käyttöliittymän tallennusverkon laitteiden hallintaan ja monitorointiin. SE:n tärkeimmät työkalut ovat: Capacity Manager, Element Manager, Event Manager, Performance Manager, Policy Manager, Provisioning Manager, Reporter ja System Manager. Capacity Manager tarjoaa graafisen näkymän tallennuslaitteiden ja palvelimien tallennuskapasiteetista. Element Managerin avulla voidaan etsiä tallennusverkon komponentteja niihin liittyvien tietojen perusteella. Event Managerin avulla voidaan selata, järjestää ja suodattaa hallittavien laitteiden lähettämiä tapahtumaviestejä. Performance Manager tarjoaa graafisia esityksiä tallennusverkon laitteiden suorituskyvystä kyseisellä hetkellä sekä menneisyydessä. Policy Managerilla voidaan automatisoida tapahtumiin reagoimista. Sen avulla voidaan esimerkiksi tehdä sääntöjä, jota SE suorittaa tietynlaisen tapahtuman ilmettyä. Provisioning Managerin avulla voidaan konfiguroida tallennusverkon laitteita. Sillä voidaan esimerkiksi tehdä uusia vyöhyke-konfiguraatioita, aliaksia tai allokoida levytilaa tallennuslaitteista. Report-työkalusta kerrotaan lisää kappaleessa 3.1.6. System Manager on työkalu, jonka avulla voidaan tarkastella yksittäisen laitteen tietoja ja laitteiden välisiä riippuvuuksia. Sillä voidaan esittää topologiakuvia koko tallennusverkosta tai sen osista. Graafisen käyttöliittymän lisäksi SE tarjoaa komentorivikäyttöliittymän, joka mahdollistaa tallennusverkon hallinnan skriptien avulla. [22, s. 2-4]

SE säilyttää keräämäänsä tietoa tietokannassa. Tietokantasovelluksena käytetään Oracle 10g Release 2 -tietokantaa. Oletuksena on, että tietokanta asennetaan samalle palvelimelle kuin Storage Essentials. [23, s. 39]

CIM-laajennukset (CIM Extensions) ovat palvelimelle asennettava agentti-ohjelmisto, joka kerää tietoa palvelimen käyttöjärjestelmästä, kuitukorteista ja niiden suorituskyvystä sekä välittää tämän tiedon SE:lle. SE:n mukana toimitetaan CIM-laajennukset seuraaville käyttöjärjestelmille: IBM AIX, HP-UX, SUSE ja Red Hat Linux, HP OpenVMS, HP Tru64 UNIX, Sun Solaris, Microsoft Windows sekä NonStop. Agenttien asentamisesta palvelimille kerrotaan lisää luvussa 4.2. [23, s. 2]

Versioon 6.1.1 asti SE:n pystyi asentamaan joko yksittäisenä sovelluksena tai integroituna HP SIM:iin. Integroidussa mallissa laitteiden löytäminen ja tapahtumien hallinta suoritettiin SIM:n avulla. Tämän tavan etu on siinä, että datakeskuksen laitteita voidaan hallita keskitetysti SIM:n kautta. SE:n uusin versio 6.2 ei tue enää integroimista SIM:n kanssa. Sen pystyy asentamaan ainoastaan yksittäisenä sovelluksena. Tallennusverkon hallinta saadaan näin eriytettyä omaan hallintasovellukseen. Muutoksen myötä kaikki tallennusverkkoihin liittyvä hallinta siirtyy SE:lle.

SE sisältää kaikki viisi toimintoa, jotka keskitetyn hallintajärjestelmän pitäisi toteuttaa. Sen avulla voidaan suorittaa automaattisesti kaikkien tallennusverkon laitteiden löytäminen (discovery). Se valvoo verkon sovellusten ja resurssien tilaa sekä tekee

automaattisia ilmoituksia mahdollisista vioista. Sillä voidaan konfiguroida tallennusverkon laitteita keskitetysti samasta sovelluksesta. SE kerää tilastotietoa suorituskyvystä, virheistä ja konfiguraatiosta, ja analysoi niitä. Tilastotietojen perusteella voidaan koostaa raportteja verkon toiminnasta. SE valvoo myös kaikkea datan hallintaan liittyviä sovelluksia ja resursseja.

3.1.2 HP Systems Insight Manager

HP SIM on ilmainen laitteistotason hallintasovellus palvelimille ja tallennuslaitteille. SIM tukee pääasiassa HP:n laitteita, mutta sillä on mahdollista hallita myös kolmannen osapuolen avoimia standardeja tukevia laitteita. Sen perustoimintoihin kuuluvat datakeskuksen laitteiden automaattinen löytäminen ja tunnistaminen, tapahtumien käsittely, hallintadatan kerääminen ja raportointi. Sovellusta voidaan myös laajentaa maksullisilla lisäosilla, jotta se tukisi tiettyjä laitetyyppejä paremmin. [25]

SIM:n hajautettu arkkitehtuuri muodostuu kolmesta osasta, jotka ovat keskitetty hallinta palvelin (CMS), hallittavat järjestelmät sekä selainta käyttävät asiakkaat. CMS ja hallittavat järjestelmät muodostavat yhdessä HP SIM-toimialueen (domain). Jokaisessa hallittavassa toimialueessa on yksi CMS, jossa HP SIM-sovellusta ajetaan. Hallintasovelluksen lisäksi CMS sisältää tietokannan, johon kaikki hallintasovelluksen keräämä tieto tallennetaan. Tietokanta voidaan sijoittaa samalle palvelimelle kuin SIM, mutta suurissa ympäristöissä se kannattaa sijoittaa omalle palvelimelle. [25, s. 37-38]

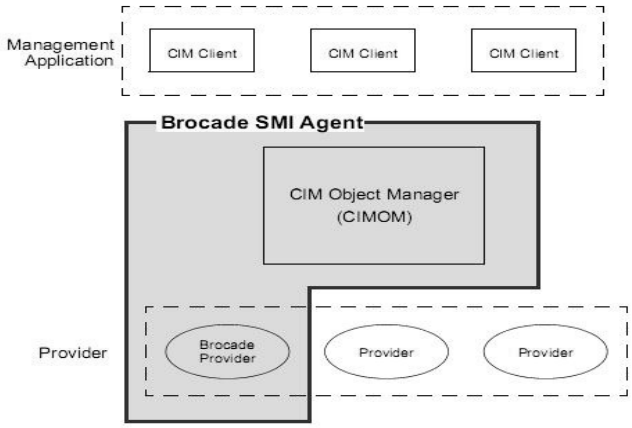
Hallittava järjestelmä on SIM:n toimialueella oleva laite, joka pystyy kommunikoimaan SIM:n kanssa jonkin protokollan avulla. Tällainen järjestelmä voi olla palvelin, pöytä- tai kannettava tietokone, tulostin, kytkin, keskitin, reititin tai tallennusverkon laite. Hallittavalla järjestelmällä pitää olla IP-osoite. Järjestelmien hallintaan voidaan käyttää useampaa kuin yhtä CMS:ää. CMS pystyy käyttämään laitteiden identifiointiin ja hallintaan montaa eri protokollaa. Mahdollisia hallintaprotokollia ovat mm. SNMP, WBEM, SMI-S ja SSH. Hallittavassa järjestelmässä pitää olla asennettuna agenttisovellus, joka vastaa CMS:n kyselyihin ja lähettää tilatietoa CMS:lle. [25, s. 38] Edellä mainituista protokollista SSH on erityisen mielenkiintoinen tämän tutkimuksen kannalta, koska se mahdollistaa komentorivikomentojen ajamisen SIM:in kautta hallittavissa laitteissa. Tähän palataan luvussa 4.2 koskien agenttien automaattista jakelua palvelimille.

HP SIM sisältää selainpohjaisen graafisen käyttöliittymän sekä komentorivikäyttöliittymän. Graafiseen käyttöliittymään tarvitaan tuettu selain, joita ovat Internet Explorer 7 ja Firefox. Komentorivikäyttöliittymää voidaan käyttää SSH:n avulla.

3.1.3 Brocade SMI Agent

Brocade on eräs tallennusverkkojen laitteita valmistava yritys. Brocade SMI Agent (SMI-A) on SMI-S agentti, joka välittää (proxy) tallennusverkon kytkimien hallintadataa SE:lle. Sen kautta voidaan hallita useita kudoksia. SMI-A asennetaan erilliselle palvelimelle, jonka käyttöjärjestelmä voi olla Sun Solaris, Microsoft Windows tai Linux. SMI-A:n avulla hallintasovellukset voivat konfiguroida ja monitoroida Brocaden tallennusverkkolaitteita yhden sovelluksen kautta. Hallittaviin laitteisiin ei tarvitse tehdä mitään muutoksia SMI-A:n käyttöönotossa vaan laitteet tukevat sitä automaattisesti. [26; 27]

SMI-A:n arkkitehtuuri muodostuu CIM:n CIMOM ja CIM Provider komponenteista (kuva 3.1). CIM asiakas, kuten Storage Essentials, kommunikoi SMI-A:n CIMOM:n kanssa, joka välittää tietoa SE:n ja CIM Providerin välillä. CIM Provider vastaavasti välittää tietoa CIMOM:n ja hallittavien järjestelmien välillä. [26, s. 2-3]



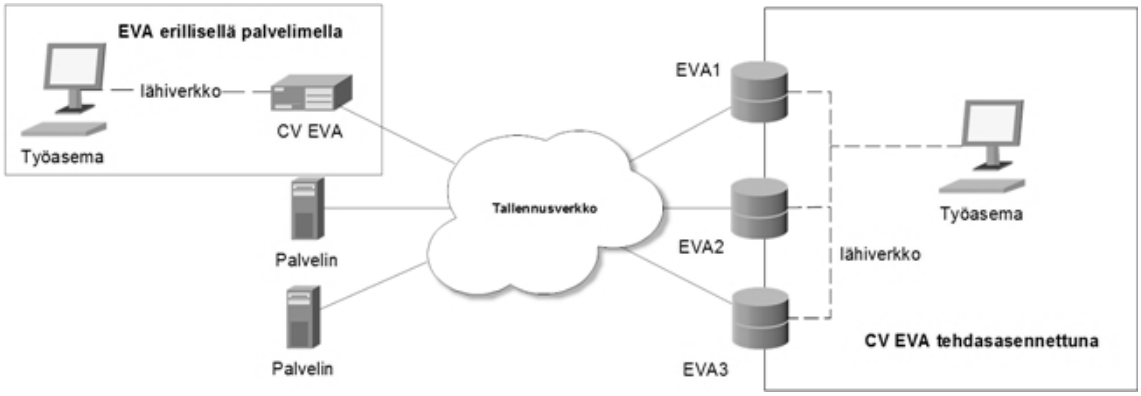
Kuva 3.1. Brocade SMI-A:n arkkitehtuuri [25]

3.1.4 HP Command View EVA

HP Command View EVA:a (CV) voidaan käyttää Enterprise Virtual Array-levyjärjestelmien konfigurointiin, hallintaan ja monitorointiin. CV tarjoaa selainpohjaisen graafisen käyttöliittymän, jonka avulla edellä mainittuja toimintoja voi käyttää. CV:hen on saatavilla myös komentorivikäyttöliittymä, jonka avulla voidaan automatisoida konfigurointia skriptien avulla. Keskitetyn hallinnan kannalta tärkein lisäosa CV:hen on SMI-S EVA, joka tarjoaa SMI-S-rajapinnan EVA-levyjärjestelmien hallintaan. SMI-S EVA:n kautta Storage Essentials pystyy hallitsemaan keskitetysti CV:n alaisuudessa olevia levyjärjestelmiä. [28, s. 13]

HP CV EVA voidaan ottaa käyttöön kahdella tavalla (kuva 3.2). Palvelinperusteisessa hallinnassa (server-based management) CV asennettaan erilliselle hallintapalvelimelle, joka on yhteydessä yhteen tai useampaan EVA-levyjärjestelmään tallennusverkon kautta. Hallintapalvelin on myös yhteydessä lähiverkkoon, jonka kautta web-

hallintakäyttöliittymään pääsee käsiksi. CV käyttää tässä tapauksessa kanavansisäistä hallintaa. [28, s. 13-14]



Kuva 3.2. HP Command View EVA voidaan asentaa erilliselle hallintapalvelimelle. EVA4400 sisältää CV:n tehdasasennettuna.

Toisessa tavassa CV on tehdasasennettuna EVA4400-levyjärjestelmässä. Tässä tapauksessa ei välttämättä tarvita erillistä hallintapalvelinta. Mikäli CV:hen halutaan liittää lisäosia kuten SMI-S EVA tai komentorivikäyttöliittymä, on ne asennettava erillisille palvelimille. Ainoastaan ohjelmiston perusosa voidaan liittää suoraan levyjärjestelmään. Jokaisen EVA4400:n pitää olla yhteydessä lähiverkkoon, jotta CV:n käyttöliittymään saadaan yhteys. Tämän tavan ongelma on se, että jokainen CV hallitsee vain yhtä levyjärjestelmää, kun taas palvelinperusteisessa tavassa pystytään hallitsemaan useita levyjärjestelmiä samasta sovelluksesta. EVA4400-levyjärjestelmiä voidaan hallinta myös keskitetysti palvelinperusteisesta hallintasovelluksesta. [28, s. 14-15]

3.1.5 HP Command View for Tape Library

HP:n hallintakortin (Interface Manager card) avulla voidaan keskittää HP:n ESL- ja EML-sarjan nauhakirjastojen hallinta yhdelle laitteelle. SE mahdollistaa nauhakirjaston komponenttien etähallinnan ja monitoroinnin. Hallintakorttia voidaan käyttää sekä graafisen käyttöliittymän että komentorivikäyttöliittymän kautta. [29, s. 13]

Hallintakorttia voidaan käyttää kolmen eri liitynnän kautta, joista kaksi on komentorivipohjaisia ja yksi graafinen. Hallintakortti sisältää sarjaportin, jolla päästään suoraan käsiksi hallintakortin komentorivikäyttöliittymään normaalin RS232-sarjakaapelin avulla. Komentorivikäyttöliittymään saadaan yhteys myös telnetin avulla. Tällöin hallintakortille on määriteltävä IP-osoite joko sarjaportin tai nauhakirjaston ohjauspaneelin (OCP) kautta. Telnetin avulla hallintakorttiin päästään käsiksi lähiverkon kautta. [29, s. 13]

HP Command View TL (CV-TL) tarjoaa graafisen selainkäyttöliittymän hallintakorttiin, jonka avulla voidaan hallita ja monitoroida nauhakirjastoja lähiverkon kautta. CV-TL:n avulla voidaan myös hallita nauhakirjastojen ohjelmistoja (firmware).

CV-TL-Sovellus asennetaan erilliselle hallintapalvelimelle, josta se kommunikoi hallintakortin kanssa lähiverkon kautta. Sitä voidaan käyttää suoraan hallintapalvelimelta tai lähiverkon kautta. [29, s. 19]

CV-TL sisältää myös SMI-S-rajapinnan, joka asennetaan CV-TL:n asennuksen yhteydessä. Tämän avulla nauhakirjastojen hallinta ja monitoirointi pystytään suorittamaan SE:n kautta.

3.1.6 HP SRM Report Optimizer

Storage Essentialsin tietokanta on erittäin laaja ja monimutkainen. Omien raporttien koostaminen suoraan tietokannasta on erittäin vaikeaa. Tätä varten on kehitetty HP SRM Report Optimizer (RO). Se on työkalu, jonka avulla voidaan koostaa raportteja SE:n tietokannasta graafisen käyttöliittymän avulla. SE:n versiosta 6.2 lähtien kaikki SE:n raportointi on siirretty RO:lle. Aikaisemmissa versioissa raportteja on voinut luoda ja selata myös SE:ssä.

RO voidaan asentaa joko samalle palvelimelle kuin SE tai omalle palvelimelle. RO kannattaa asentaa omalle palvelimelle, koska sekä SE että RO ovat erittäin raskaita sovelluksia, joilla molemmilla on omat tietokantasovellukset. Eri palvelimille asentamista tukee myös se, että yhden palvelimen hajoaminen ei lopeta molempien sovelluksien suoritusta. Toimiakseen RO vaatii kolme komponenttia: raporttisovelluksen ja raporttitietokannan sekä Oracle-tietokannan asiakassovelluksen [30]. Oracle-asiakassovellus tarvitaan, jotta RO pääsee käsiksi SE:n tietokantaan. Raporttisovellus on rakennettu SAP BusinessObjects XI 3.1-sovelluksen päälle. BusinessObjects on yrityksille tarkoitettu suorituskyvyn, suunnittelun, raportoinnin ja analysoinnin hallintatyökalu. Sen avulla voidaan koostaa SQL-tietokannasta dokumentteja graafisen käyttöliittymän avulla ilman, että dokumentin laatijan ymmärtää relaatiotietokannan toimintaa. Report Optimizer on rakennettu lisäosaksi, BusinessObjects XI:hin.

RO ei käytä suoraan SE:n tietokantaa raporttien koostamiseen, vaan sillä on oma tietokantasovellus, johon se lataa tarvitsemansa tiedot käyttäen Oraclen asiakassovellusta. Raporttitietokannan päivitystaajuuden voi määritellä SE:n asetuksista. (Configurations → Reports). RO:n tietokantasovelluksena käytetään MySQL:ää. [30]

3.2 Hallinta-palvelimet

Tallennusverkon hallintasovellukset on hajautettu kolmelle eri palvelimelle. Jokainen palvelin on oma fyysinen laitteensa. Virtuaalikoneita ei käytetä tämän ympäristön hallintapalvelimissa, koska hallintasovelluksilla on melko korkeat suorituskykyvaatimukset. Hallintasovellukset on jaettu seuraavasti palvelimien kesken:

- **Asolia**: HP Storage Essentials SRM ja Systems Insight Manager

- **Emaster**: HP Command View EVA ja Brocade SMI Agent
- **Villey**: HP SRM Report Optimizer ja HP Command View TL.

Asolia, Emaster ja Villey ovat kyseisten palvelimien nimiä.

Asolia on HP Proliant BL685c G1 korttipalvelin, jossa on neljä AMD Opteron 8220 (2.8 GHz) -prosessoria sekä 32 GB keskusmuistia. Tallennustilaa palvelimesta löytyy 272 GB. Käyttöjärjestelmänä toimii Windows Server 2008 64bit SP1 Enterprise Edition. SE:n ja SIM:n laitteistovaatimukset löytyvät taulukosta 3.2. Asolian laitteisto riittää hyvin sekä SE:n että SIM:n ajamiseen samalla koneella. SE:stä käytetään uusinta versiota 6.2.0 ja SIM:stä versiota 5.3.1.

Emaster on HP DL380 G4 Storage-palvelin, jossa on neljä Inter Xeon (3,4GHz) yksiytimistä prosessoria. Keskusmuistia Emasterissa on neljä gigatavua ja tallennustilaa 100 GB. Palvelimen käyttöjärjestelmä on Windows Server 2003 32bit SP2 Enterprise Edition. CV:n ja SMI-A:n laitteistovaatimukset eivät ole niin kovia kuin SE:n, joten näitä kahta voidaan ajaa vanhemmalla palvelimella. SMI-A:n muistivaatimus riippuu paljon tallennusverkossa olevien porttien määrästä. 2 GB riittää hyvin alle 5000 portin tallennusverkkoon. Näistä ohjelmistoista käytetään versioita 9.01 (CV) ja 120.10.0 (SMI-A). Kyseistä SMI-A:n versiota käytettäessä kytkimien Fabric OS:n versio tulisi olla vähintään 6.3.x.

RO:n laitteistovaatimukset ovat kovemmat kuin CV:llä ja SMI-A:lla, joten sitä ajetaan CV-TL:n kanssa omalla palvelimellaan. Villey on HP BL460c G1 korttipalvelin, jossa on kaksi Intel Xeon 5270 tuplaydinprosessoria (3,5GHz). Keskusmuistin määrä on 32 GB ja tallennustilaa on 72 GB. Palvelimen käyttöjärjestelmänä toimii Windows Server 2003 SP2 32bit, koska RO:n edellinen versio ei tukenut vielä 64-bittistä Windows Server 2008 käyttöjärjestelmää. Tämän hetkinen versio 6.2.0 tukee myös sitä. RO:n dokumentaation mukaan se vaatisi vähintään 300 GB tallennustilaa. Palvelimen 72 GB:n kiintolevy riittää kuitenkin hyvin kyseisessä ympäristössä.

Taulukossa 3.1 on listattuna kaikkien hallintapalvelimien nimi, valmistaja, malli, prosessori(t), muistimäärä, tallennnustilan määrä sekä käyttöjärjestelmä

Taulukko 3.1. Hallintaan käytettävien palvelimien nimi, valmistaja, malli, prosessori(t), keskusmuistin määrä, tallennustila ja käyttöjärjestelmä

Nimi	Valmistaja	Malli	CPU	Muisti	Kiintolevy	Käyttöjärjestelmä
Asolia	HP	BL685c G1	4x Inter Xeon 8220 2.8GHz	32GB	370GB	Windows Server 2008 Enterprise Edition SP1 64bit
Emaster	HP	DL380 G4	4 x Inter Xeon 3,4 GHz	4GB	100GB	Windows Server 2003 Enterprise Edition SP2 32bit
Villey	HP	BL460c G1	2 x Inter Xeon 5270 3,5GHz	32GB	72GB	Windows Server 2003 Enterprise Edition SP2 32bit

Taulukossa 3.2 on esitelty hallintaohjelmistojen laitteistovaatimukset asennetuille ohjelmistoversioille. Kyseiset ohjelmistoversiot ovat kaikki SE:n tukemia.

Taulukko 3.2. Hallintasovellusten järjestelmävaatimukset

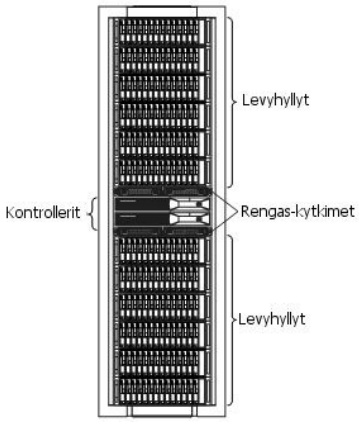
Ohjelma	Prosessori	Muisti	Kiintolevy	Käyttöjärjestelmä	Asennettu Versio	Muuta
HP Storage Essentials SRM	neliydin	12GB	200GB	Windows Server 2003 SP tai R2 SP2 32bit, Windows 2008 SP1 tai SP2 64bit	v6.2.0	
HP Systems Insinght Manager	2,4GHz	2GB	-	Windows Server 2003, 2008 ja Windows XP ja Vista	v5.3.1	
HP Command View EVA	1,26GHz	2GB	2GB	Windows Server 2003 tai 2008	v9.01	
Brocade SMI Agent	3GHz	2GB	-	Windows Server 2003 SP2, 2008, Vista Business (vain 32bit)	v120.10.0	Fabric OS 6.3.x kytkimissä
HP SRM Report Optimizer	neliydin 2,33GHz	12GB	300GB	Windows Server 2003 SP tai R2 SP2 32bit, Windows 2008 SP1 tai SP2 64bit	v6.2.0	
HP Command View TL	Pentium IV 1.6GHz	512MB	-	Windows 2000 Pro tai Server SP3, Windows Server 2003, Windows XP	v2.3	

3.3 Tallennusverkon laitteet

Tässä kappaleessa kerrotaan lyhyesti laitteista, joita tämän diplomityön kohteena olevassa tallennusverkossa on. Tallennusverkko koostuu pääasiassa HP:n levyjärjestelmistä ja palvelimista sekä Brocaden valmistamista tallennusverkon kytkimistä.

3.3.1 HP Enterprise Virtual Array

HP EVA on modulaarinen levyjärjestelmä. Modulaarisuudella tarkoitetaan, sitä että levyjärjestelmä koostuu erillisistä moduuleista (levyhyllyistä), joita voidaan tarvittaessa lisätä levyjärjestelmän kapasiteetin kasvattamiseksi (kuva 3.3). EVA:t jaetaan 4000-, 6000- ja 8000-sarjan levyjärjestelmiin. Eri sarjat eroavat toisistaan kiintolevyjen maksimimäärässä, maksimitallennuskapasiteetissa, välimuistin määrässä, suorituskyvyssä ja kontrollerien älykkyydessä. [31, s. 21-23]



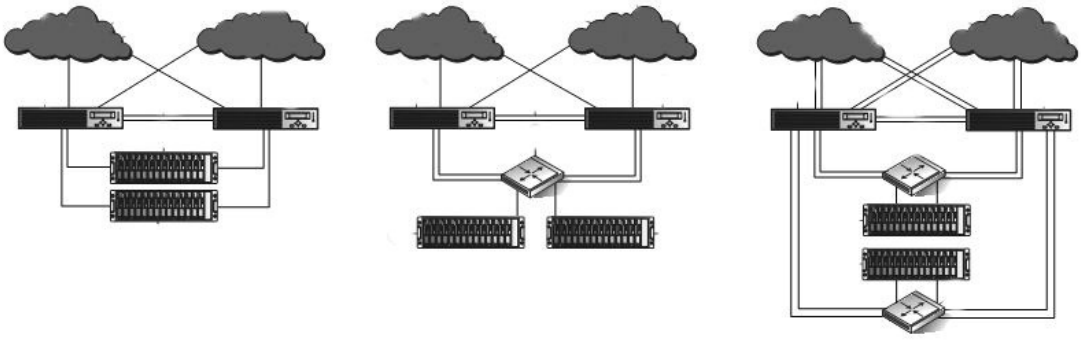
Kuva 3.3. EVA-levyjärjestelmä

EVA-levyjärjestelmän laitteisto koostuu kontrollereista, rengaskytkimistä (Loop Switch) ja levyhyllyistä (Drive Enclosure). EVA4000- ja EVA6000-levyjärjestelmät sisältävät HSV200-kontrollerin ja EVA8000 HSV210-kontrollerin (kuva 3.4). Jokainen levyjärjestelmä sisältää aina kaksi kontrolleria, jotka ovat redundanttisia keskenään. Yksi HSV200-kontrolleri sisältää kaksi kahden gigabitin kuituporttia, ja HSV210 neljä neljän gigabitin kuituporttia. Välimuistia HSV200:ssa on yksi gigatavu ja HSV210:ssa kaksi gigatavua. HSV200:aan voidaan lisätä 112 kiintolevyä ja HSV210:iin 240. [31, s. 80]

Levyhyllyt ovat kehikoita, joihin fyysiset kiintolevyt sijoitetaan. Yksi levyhylly on 3 U:n (1U = 44,45 mm) korkuinen ja siihen voidaan lisätä maksimissaan 14 kiintolevyä. Levyhyllyt yhdistetään kahden gigabitin kuitukytkennöillä rengaskytkimien kautta tai suoraan molempiin kontrollereihin.

Rengaskytkimet ovat levyjärjestelmän keskitetty liityntäpiste, joka yhdistää kontrollerit ja levyhyllyt toisiinsa. Mikäli levyjärjestelmässä on yli neljä levyhyllyä, täytyy levyjärjestelmään lisätä rengaskytkimet. EVA6000-levyjärjestelmässä käytetään kahta ja EVA8000-levyjärjestelmässä neljää rengaskytkintä levyhyllyjen liittämiseen.

EVA8000-levyjärjestelmä liitetään kudokseen käyttämällä kontrollerien neljää kuituporttia. Molemmista kontrollereista voidaan viedä kaksi kuitua molempiin kudoksiin. Kontrollerit yhdistetään levyhyllyihin kahden rengasparin kautta (neljä rengaskytkintä). EVA4000- ja 6000-levyjärjestelmän kontrollerit liitetään molempiin kudoksiin ainoastaan yhdellä kuidulla, koska yksi kontrolleri sisältää ainoastaan kaksi porttia. EVA6000:ssa kontrollerit liittyvät levyhyllyihin yhden rengasparin kautta (kaksi rengaskytkintä). EVA4000:ssa levyhyllyt ja kontrollerit muodostavat keskenään renkaan ilman rengaskytkimiä. [31, s. 29-32]



Kuva 3.4. EVA4000- (vas.), EVA6000- (kesk.) ja EVA8000-sarjan (oik.) levyjärjestelmien komponenttien väliset liitynnät [29, s. 29-32]

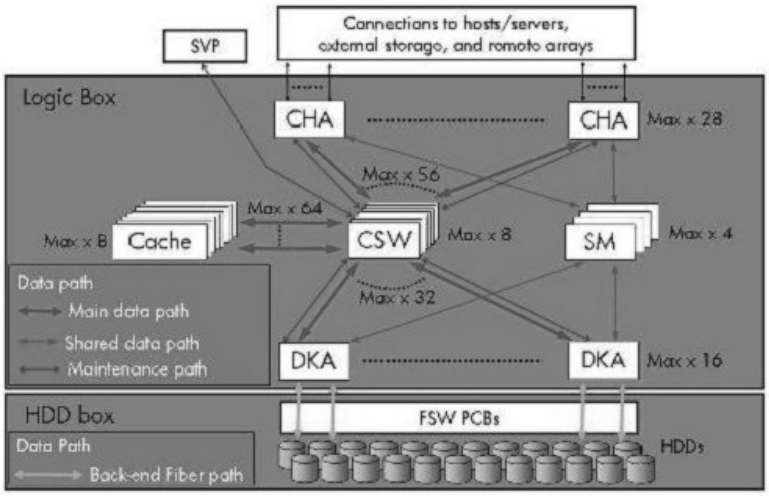
Kehitettävä ympäristö sisältää kaksi EVA4000- ja kuusi EVA6000-levyjärjestelmää sekä yhden EVA8000-levyjärjestelmän. Levyjärjestelmien tiedot löytyvät alla olevasta taulukosta. EVA-levyjärjestelmien kytkennät on selitetty kohdassa 3.2.4.

3.3.2 HP XP24000

HP XP24000 on suuryritysten levyjärjestelmä, joka on tarkoitettu sovelluksille, joissa ei saa ilmetä käyttökatkoksia. XP:n kaikki komponentit on vähintään kahdennettu, jotta järjestelmässä ei ole komponentteja, joiden hajoaminen vaikuttaisi levyjärjestelmän toimintaan. XP-levyjärjestelmä tarjoaa korkeaa luotettavuutta, skaalautuvuutta ja suorituskykyä. XP-levyjärjestelmään voidaan kytkeä maksimissaan 224 kuituporttia, 4-512 gigatavua välimuistia ja maksimissaan 1152 kiintolevyä. [32, s. 7-9]

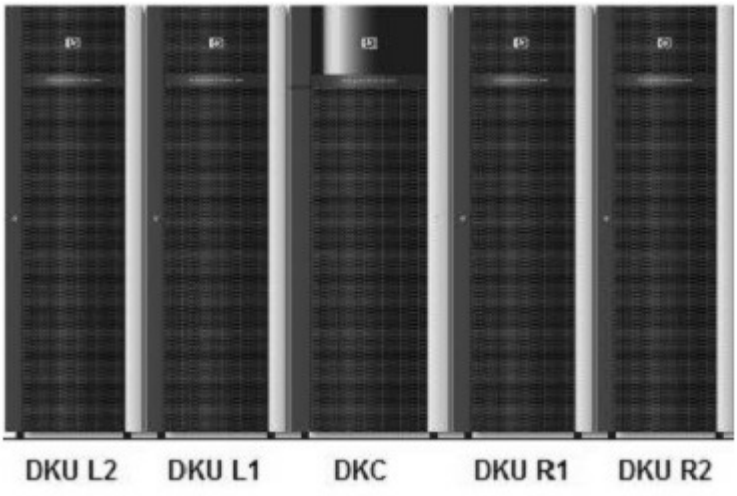
XP24000-levyjärjestelmän arkkitehtuuri koostuu yhdestä DKC-kabinetista (Disk Control Frame) ja DKU-laajennuskabineteista (Disk Array Frame). DKC-kabinetti on levyjärjestelmän keskus, joka sisältää logiikkakehikon (Logic Box) ja kiintolevykehikon (HHD Box) (kuva 3.5). Logiikkakehikko koostuu kanavasovittimista (CHA), kiintolevysovittimista (DKA), välimuistisovittimista (CMA) ja jaetun muistin sovittimista (SMA). [32, s. 21] DKC on jaettu kahteen itsenäiseen ja redundanttiseen klusteriin. Ensisijainen klusteri sijaitsee DKC-kabinetin etupuolella ja toissijainen sen takapuolella. Mikäli jompikumpi klusteri hajoaa, toinen pystyy jatkamaan levyjärjestelmän toimintaa itsenäisesti. [32, s. 24]

DKC sisältää myös hallintakonsolin (SVP, Service Prosessor), jonka kautta levyjärjestelmää voidaan konfiguroida ja monitoroida. SVP kerää myös tietoa XP:n suorituskyvystä diagnosointia ja analysointia varten. SVP:n kautta voidaan päivittää levyjärjestelmän ohjemistoa. [32, s. 25] Jotta XP-levyjärjestelmää voidaan hallita keskitetysti Storage Essentialin avulla, täytyy SE:llä olla pääsy XP:n SVP:hen.



Kuva 3.5. XP24000-levyjärjestelmän DKC-kabinetin arkkitehtuuri

DKC:n kiintolevykehikkoon voidaan lisätä maksimissaan 128 kiintolevyä. Tämän lisäksi XP-levyjärjestelmää voidaan laajentaa DKU-laajennuskabineteilla. Laajennuskabinetteja voidaan lisätä maksimissään neljä kappaletta. Laajennuskaapit sijoitetaan DKC:n molemmille puolille ja niiden tunnisteet ovat DKU L1, DKU L2, DKU R1 ja DKU R2 (kuva 3.6). Jokaiseen laajennuskaappiin voidaan lisätä maksimissaan 256 kiintolevyä, josta muodostuu XP-levyjärjestelmän maksimilevymäärä 1152 (4 x 256 + 128 = 1152). [32, s. 23]



Kuva 3.6. XP24000-levyjärjestelmän laajennuskabinetit

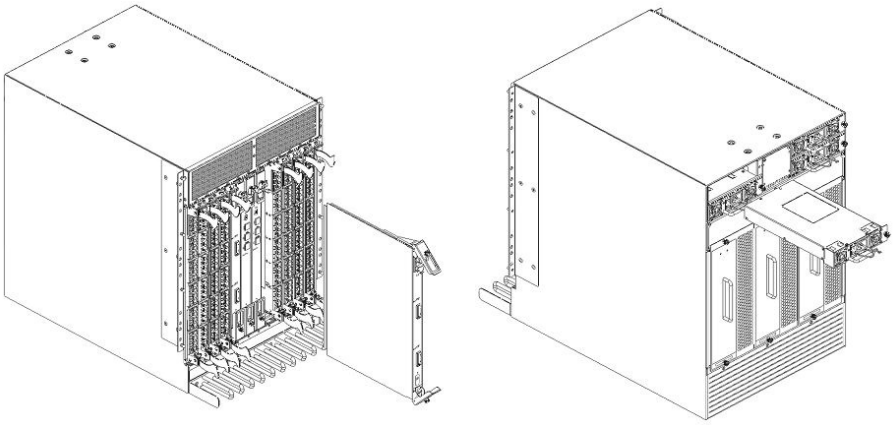
XP-levyjärjestelmään voidaan lisätä useita erikokoisia ja -tyyppisiä levyjä. Tuettuja levytyyppejä ovat mm. FATA, SATA ja SSD-kiintolevyt. Osa XP24000:n kiintolevyistä voidaan määritellä varalevyiksi, joista yksi otetaan käyttöön jonkin käytössä olevan levyn hajotessa. Varalevy voi korvata minkä tahansa levyn mistä tahansa DKU:sta kunhan levyjen pyörimisnopeudet ovat samat ja varalevyn kapasiteetti on vähintään yhtä suuri tai suurempi kuin hajonneen levyn. Varalevyjä voi olla maksimissaan 40 kappaletta. [32, s. 25]

XP-levyjärjestelmän yksi hyödyllisimpiä ominaisuuksia on Thin Provisioning-tekniikka (THP). Sen avulla palvelimille voidaan allokoida enemmän tallennustilaa, mitä levyjärjestelmän fyysinen kapasiteetti todellisuudessa on. Jos palvelimelle allokoidaan 500 GB:n looginen kiintolevy, XP ei THP:tä käyttämällä varaa tätä tallennustilaa heti kokonaan, vaan varaa sitä ainoastaan sitä mukaan kun palvelin kirjoittaa dataa kiintolevylle. Palvelimelle kiintolevy näkyy kuitenkin kokoajan 500 gigatavuna. Näin voidaan maksimoida kaiken fyysisen tallennuskapasiteetin tehokas käyttö. [32, s. 10]

3.3.3 Brocade Silkworm 48000

Brocade Silkworm 48000 (BS48000) on director-luokan tallennusverkon kytkin. Se tarjoaa korkeaa luotettavuutta, saatavuutta ja suorituskykyä tallennusverkon kriittisimpiin kohtiin. BS48000 on modulaarinen kytkin, jota voidaan laajentaa uusille porttikorteilla (Port Blade) tallennusverkon laitemäärän kasvaessa. Siihen voidaan lisätä maksimissaan 384 kuituporttia, joiden nopeudet voivat olla 1, 2, 4, 8 tai 10 Gbps. Portteja voidaan myös niputtaa 2-8 portin kokonaisuuksiksi muodostaen maksimissaan 64 Gbps ISL-linkin. [33, s. 19] Kehikon sisäinen siirtonopeus on 3 Tbit/s ja se takaa 99,999 % saatavuuden.

BS48000:n modulaarinen rakenne koostuu hallintakorteista, porttikorteista, tuuletin- ja teholähdekomponenteista sekä WWN-korteista (kuva 3.7). Hallinta- ja porttikortit sijaitsevat kehikon etupuolella, tuuletin- ja teholähdekomponentit sekä WWN-kortit kehikon takapuolella. Hallintakortteja voidaan lisätä kaksi kappaletta BS48000:aan. Hallintakorteista ainoastaan toinen on aktiivinen ja se hallitsee kaikkia kehikkoon liitettyjä portteja. Toinen hallintakortti odottaa valmiustilassa. Jos aktiivinen hallintakortti hajoaa, siirtyy kehikon hallinta valmiustilassa odottavalle kortille. BS48000:n porttimäärää voidaan kasvattaa lisäämällä siihen uusia porttikortteja. Porttikortteja voidaan lisätä maksimissaan kahdeksan kappaletta, joissa yhdessä voi olla joko 16, 32, tai 48 kahdeksan gigabitin porttia. Lisäksi kehikkoon voidaan lisätä kymmenen gigabitin porttikortti, jossa on kuusi porttia. Tuulettimet ja teholähteet voidaan vaihtaa ajon aikana, ilman että se vaikuttaa kytkimen toimintaan. Kahden redundanttisen WWN-kortin tehtävä on säilyttää kehikkoon liittyvää informaatiota, kuten WWN- ja IP-osoitteita, sekä tilatietoa jokaisesta kortista sekä teholähteestä. [33, s. 20]



Kuva 3.7. Brocade Silkworm 48000 director-luokan kytkin edestä ja takaa

Kytkimen älykkyydestä vastaa sen käyttöjärjestelmä, jonka nimi on Fabric OS (FOS). FOS koostuu sovelluksista, joita ajetaan sulautetun Linux-ytimen päällä. Näitä sovelluksia ovat mm. nimi-, alias-, vyöhyke- ja aikapalvelin sekä hallintarajapinnat, kuten SNMP-agentti sekä SMI-S-sovellusrajapinta. [33, s. 25] FOS:n käyttö tapahtuu, joko käyttämällä komentorivikäyttöliittymää, graafisia työkaluja tai hallintarajapintojen avulla. Komentorivikäyttöliittymään päästään käsiksi joko sarjaporttiliitynnällä tai ottamalla yhteys kytkimen hallintakortin IP-osoitteeseen Telnetin tai SSH:n avulla. Graafisen käyttöliittymän FOS:iin tarjoaa kytkimen oma selainkäyttöliittymä (Web Tools) ja erillinen hallintaohjelmisto Data Center Fabric Manager (DCFM).

Kehitettävän tallennusverkon ytimen muodostavat kuusi Brocade Silkworm 48000 director-kytkintä. Jokaisen kytkimen Fabric OS:n versio on tämän diplomityön kirjoitushetkellä 6.3.0a. Taulukossa 3.3 on lueteltu kaikkien kytkimien nimet, valmistaja, malli, FOS-versio, toimialueen tunniste (dynaamisen osoitteen domain-osa), kytkettyjen porttien määrä sekä porttien yhteismäärä. Kytkimien muodostama topologia on selitetty kohdassa 3.4.

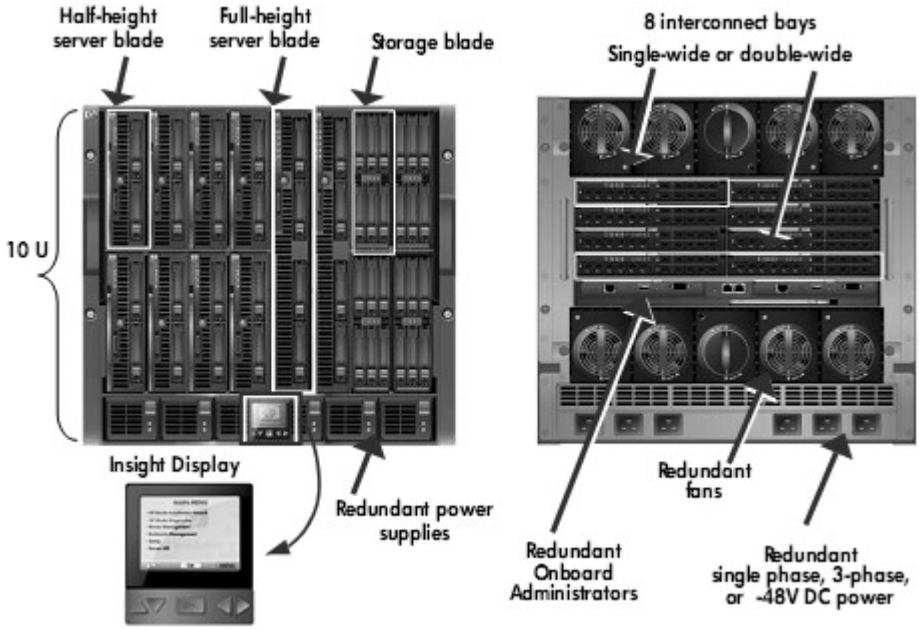
Taulukko 3.3. Kehitettävän ympäristön Brocade Silkworm 48000 kytkinten tiedot

Nimi	Valmistaja	Malli	FOS versio	Kudoksen nimi	Toimialueen tunniste	Kytkettyjä portteja	Porttien lkm
Xenon	Brocade	Silkworm 48000	v6.3.0a	x86_A	105	136	160
Crypton	Brocade	Silkworm 48000	v6.3.0a	x86_B	106	132	159
Chlorine	Brocade	Silkworm 48000	v6.3.0a	x86_A	15	92	140
Iodine	Brocade	Silkworm 48000	v6.3.0a	x86_B	16	93	140
Radium	Brocade	Silkworm 48000	v6.3.0a	x86_A	205	41	49
Barium	Brocade	Silkworm 48000	v6.3.0a	x86_B	206	41	48

3.3.4 Palvelimet

Tallennusverkkoon kytkettyjä fyysisiä palvelimia on noin 900 kappaletta. Valtaosa fyysisistä palvelimista on HP:n BL-sarjan korttipalvelimia. Fyysisten palvelimien lisäksi tallennusverkkoon on kytketty useita satoja virtuaalikoneita, joiden kiintolevyt sijaitsevat tallennusverkossa. Virtuaalikoneita ajetaan pääasiassa HP DL585- sekä BL490c-palvelimilla VMware ESX-virtualisointisovelluksen päällä.

BL-sarjan korttipalvelimet sijoitetaan palvelinkehikkoon, joka vastaa palvelimien keskitetystä tehotarpeesta, ilmanvaihdosta, etähallinnasta ja I/O-liitynnöistä. Korttipalvelimien tarkoitus on vähentää datakeskuksen kustannuksia keskittämällä palvelimien tarpeet kehikolle ja vähentämällä palvelimien käyttämää lattiapinta-alaa. [35, s. 4] HP BladeSystem c7000 palvelinkehikkoon voidaan lisätä kahdeksan täyskorkeaa (full height), 16 puolikorkeaa (half height) ja 32 tuplatiheys (double density) korttipalvelinta [24].



Kuva 3.8. HP BladeSystem c7000 korttipalvelinkehikko [33]

BladeSystem c7000 kehikko on rakennettu erillisesti moduuleista (kuva 3.8), jotka ovat redundanttisia keskenään. Kehikon etupuolelle sijoitetaan palvelimet sekä teholähteet. Teholähteitä voidaan liittää kehikkoon kuusi kappaletta ja ne voidaan vaihtaa ajon aikana ilman, että kehikkoa tai palvelimia täytyy sammuttaa. Kehikon taakse kytketään tuulettimet, liityntämoduulit (interconnects) ja hallintamoduulit (Onboard Administrator). Tuulettimia voidaan liittää kehikkoon kymmenen kappaletta ja ne voidaan myös vaihtaa ajon aikana. Kehikkoon voidaan liittää maksimissaan kahdeksan liityntämoduulia. Vierekkäiset liityntämoduulit ovat redundanttisia keskenään. Keskitetyt liityntämoduulit vähentävät huomattavasti kaapeloinnin tarvetta datakeskuksessa, koska jokaiselle palvelimelle ei tarvitse kytkeä omia kaapeleita

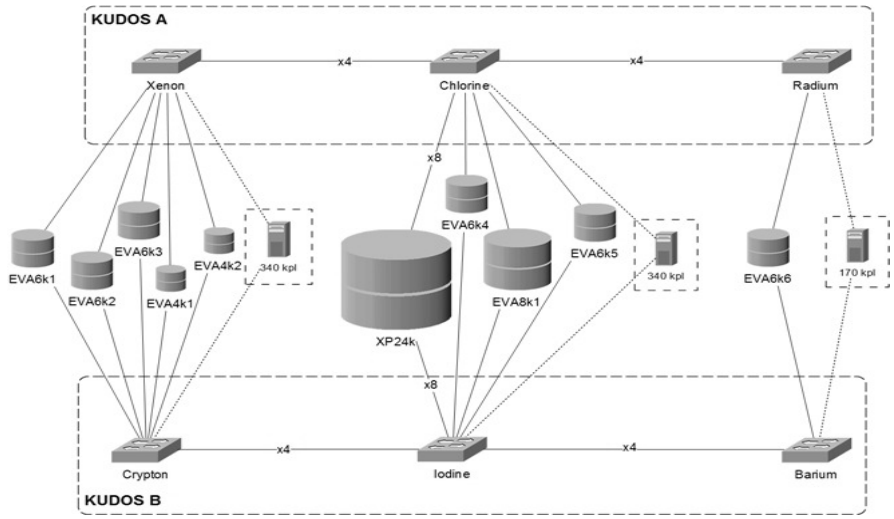
lähiverkkoa, tallennusverkkoa ja etähallintaa varten. Tämä säästää portteja lähiverkon ja tallennusverkon kytkimissä. Hallintamoduulit tarjoavat keskitetyn etähallintayhteyden kaikkiin kehikon palvelimiin. Niitä voidaan liittää kaksi kappaletta yhteen kehikkoon. [35, s. 5]

HP DL585 on 3U:n korkuinen tavallinen laitehyllyyn sijoitettava palvelin, johon voidaan liittää neljä prosessoria ja satoja gigatavuja muistia. Laajennuskorteille siihen voidaan liittää useita lähiverkko- ja kuituliityntöjä. Tästä syystä se sopii hyvin virtuaalikoneita ajavaksi fyysiseksi palvelimeksi.

HP BL490c G6 on puolikorkea korttipalvelin, joka on erityisesti suunniteltu virtuaalikoneita ajavaksi palvelimeksi. Pienestä koosta huolimatta, siinä on 18 kappaletta DDR3-muistipaikkoja ja se tukee myös 10 gigabitin Ethernetiä. [36]

3.4 Topologia

Kehitettävä tallennusverkko (kuva 3.9) koostuu kahdesta erillisestä kudoksesta, joiden runko muodostuu kolmesta tallennusverkon kytkimestä. Molempien kudoksien topologia on kolmen kytkimen muodostama kaskadi ja kudokset ovat redundanttisia keskenään. Kytkimet muodostavat kolme paria, jotka ovat Xenon-Crypton, Chlorine-Iodine sekä Radium-Barium. Jokainen pari on sijoitettu omaan laitetilaansa tarjoten pääsyn kumpaankin kudokseen kyseisestä tilasta. Molemmissa kudoksissa kytkimet on yhdistetty toisiinsa nelinkertaisilla ISL-linkeillä (trunk), jotta kuorma voidaan jakaa usealle fyysiselle kuitukaapelille ja että yhden kuidun vioittuminen ei katkaise yhteyttä kytkimien välillä. Suurin osa verkon kuitukaapeleista on 62,5 µm:n monimuotokuituja, mutta ISL-linkit kytkimien Chlorine ja Radium sekä Iodine ja Barium ovat 9 µm:n yksimuotokuituja, koska näiden kytkimien etäisyys toisistaan on hieman pidempi. Kaikki kytkimet ovat Brocaden valmistamia director-luokan kytkimiä, joiden malli on Silkworm 48000.



Kuva 3.9. Kehitettävän tallennusverkon topologia

Periaatteena voidaan pitää sitä, että kaikki tallennusverkon päätelaitteet on kytketty molempiin kudoksiin yhdellä tai useammalla kuitukaapelilla. Kytkinparit Xenon-Crypton ja Chlorine-Iodine muodostavat tallennusverkon vanhemman osan, josta johtuen niihin on kytkettynä tällä hetkellä eniten palvelimia ja tallennuslaitteita. Tallennusverkkoa on laajennettu hiljattain uuteen laitetilaan, jossa sijaitsevat kytkimet Radium ja Barium.

Kytkimiin Xenon ja Crypton on kytketty levyjärjestelmät EVA4k1 ja EVA4k2 sekä EVA6k1-EVA6k3. Levyjärjestelmien nimet kuvaavat niiden mallia. Levyjärjestelmät EVA4kX ovat HP:n valmistamia EVA4000-levyjärjestelmiä sekä EVA6kX ovat vastaavasti EVA6000-levyjärjestelmiä. Levyjärjestelmien lisäksi kytkimiin Xenon ja Crypton on kytkettynä noin 340 palvelinta.

Topologian keskellä oleviin kytkimiin Crypton ja Iodine on kytketty kaksi EVA6000-levyjärjestelmää, EVA6k4 ja EVA6k5. Näiden lisäksi niihin on kytketty EVA8000-sarjan levyjärjestelmä EVA8k1 sekä XP24000-sarjan levyjärjestelmä XP24k. Myös tähän kytkinpariin on kytketty noin 340 palvelinta.

Kytkimiin Radium ja Barium on kytketty ainoastaan levyjärjestelmä EVA6k6 sekä noin 170 palvelinta. Tallennusverkon laajetessa uudet päätelaitteet tullaan kytkemään näihin kytkimiin, koska laitetilat, joihin muut kytkimet on sijoitettu, ovat melko täynnä.

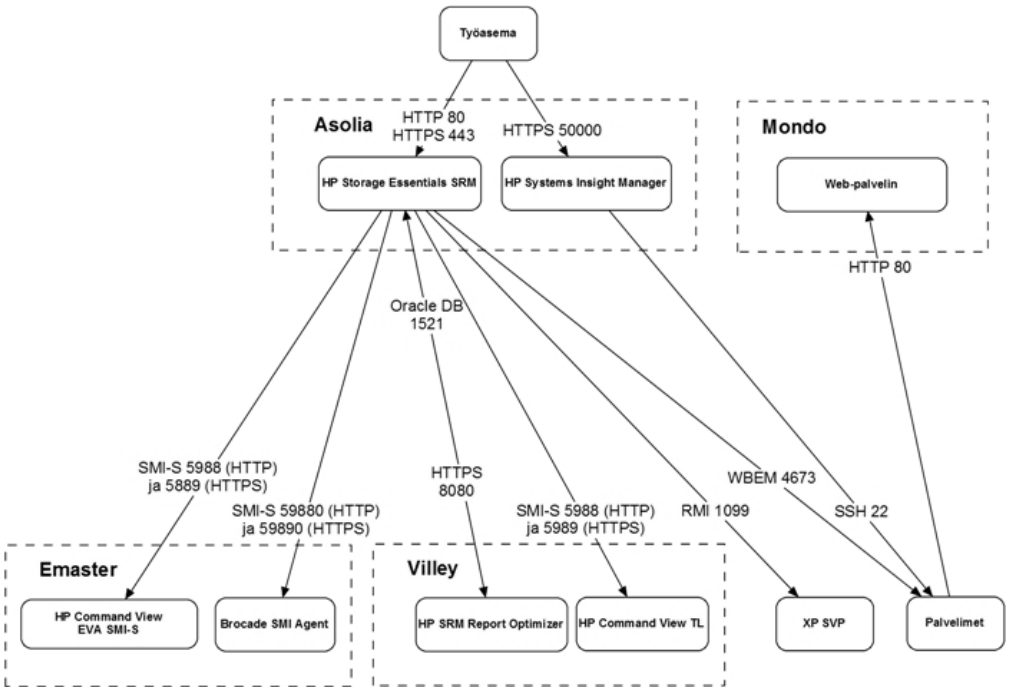
Suurin osa datakeskuksen palvelimista on korttipalvelimia (blade-server), jotka sijotetaan palvelinkehikkoon. Kehikko sisältää tallennusverkon kytkimen tai vastaavan laitteen, joka kerää kehikon palvelimien liikenteen ja ohjaa sen tallennusverkkoon. Yhdessä kehikossa on kaksi tällaista moduulia, joista viedään kaksi kuitukaapelia molempiin kudoksiin. Yksi palvelin ei siis varaa aina yhtä porttia runkoverkon kytkimestä.

4 TULOKSET

Tässä luvussa käsitellään tämän diplomityön tekemisen yhteydessä syntyneet tulokset, joita ovat tallennusverkon keskitetty hallinta ja monitorointi sekä SE:n agenttien automaattinen jakelujärjestelmä.

4.1 Tallennusverkon keskitetty hallinta ja monitorointi

Tallennusverkon hallintaan ja monitorointiin käytetään sovelluksia, jotka on kuvattu luvussa 3.1. Kuvassa 4.1 on kuvattuina eri hallintasovellusten ja -palvelimien väliset riippuvuudet. Katkoviivalla piirretyt laatikot kuvaavat fyysistä palvelinta, ja pyöristetyillä kulmilla piirretyt laatikot hallintaan käytettäviä sovelluksia. Sovellusten väliset nuolien suunnat kuvaavat aktiivisen osapuolen yhteydenottoa. Nuolen kärkeä lähinnä oleva teksti kuvaa yhteydenoton kohteessa käytettävää TCP-porttia. Tekstikentässä on myös sovellusten välillä käytettävä protokolla.



Kuva 4.1. Hallintasovellusten käyttämät TCP-portit

SE:n graafinen selainkäyttöliittymä käyttää HTTP:n ja HTTPS:n oletusportteja 80 ja 443. SIMin selainkäyttöliittymä käyttää HTTPS-protokollaa ja porttia 50000. SE kommunikoi CV:n ja SMI-A:n kanssa SMI-S hallintaprotokollan avulla. SMI-S käyttää oletuksena TCP-porttia 5988 HTTP-yhteyksille ja porttia 5989 HTTPS-yhteyksille.

Koska *Emaster* sisältää kaksi SMI-S:ää käyttävää sovellusta, täytyy toisen sovelluksen käyttämät TCP-portit muuttaa. Tässä tapauksessa SMI-A:n käyttämät portit on vaihdettu portteihin 59880 (HTTP) ja 59890 (HTTPS).

SE ja CV-TL käyttävät kommunikointiin myös SMI-S:ää. CV-TL:n päässä käytössä ovat oletusportit 5988 ja 5989. Report Optimizeriin SE:stä pääsee käsiksi käyttämällä HTTPS-protokollaa ja TCP-porttia 8080. RO hakee raporttitietokantaansa tietoa SE:n Oracle-tietokannasta. Oracle käyttää oletuksena TCP-porttia 1521. Oletusporttia ei ole muutettu.

XP-levyjärjestelmän hallintaprosessorin SVP:n ja SE:n välillä käytetään Java RMI-sovellusrajapintaa (Remote Method Invocation). RMI-rajapintaan saadaan yhteys TCP-portin 1099 kautta. Mikäli käytössä olisi HP Command View XP, voitaisiin XP-levyjärjestelmän hallintaan käyttää myös SMI-S-protokollaa.

Hallittaville palvelimille lisättävät SE:n CIM-laajennukset käyttävät oletusporttia 4673. SE:n agenttien lisäksi palvelimien hallintaan käytetään SSH:ta. SIM käyttää SSH:n oletusporttia 22 agenttien asentamiseksi ja testaamiseksi. Hallittava palvelin lataa HTTP-protokollan avulla agenttien asennustiedostot web-palvelimelta (Mondo). Web-palvelimen TCP-portti on 80.

4.1.1 Ajastetut toiminnot

Jotta tallennusverkon keskitetty hallinta ja monitorointi pysyy ajan tasalla, pitää useita toimenpiteitä ajastaa toimimaan automaattisesti tietyn väliajoin. Tällaisia toimenpiteita ovat mm. SIM:n laitteiden löytäminen, SE:hen lisättyjen laitteiden yksityiskohtaisten tietojen kerääminen, agenttien automaattinen lisäys hallittaville palvelimille, raporttitietokannan päivitys sekä suorituskykydatan kerääminen kaikista hallittavista kohteista. Kaikkia toimintoja ei ole hyvä suorittaa samanaikaisesti, jotta palvelimet eivät kuormitu liikaa. Lisäksi toiminnot kannattaa ajastaa suoritettavaksi silloin kun datakeskuksen kuormitusaste on alhainen.

Taulukoissa 4.2 ja 4.3 on kuvattuna ajastettujen toimenpiteiden sijoittuminen vuorokauteen viikon eri päivinä. Taulukossa 4.2 ovat ajastetut toiminnot maanantaista perjantaihin ja taulukossa 4.3 lauantaista sunnuntaihin. Taulukkojen x-akselilla on kuvattuna yksi tunti viiden minuutin välein ja y-akselilla vuorokauden tunnit. Taulukossa 4.1 on lueteltuna toimintojen värikoodit, kirjoitustunnisteet ja miten usein toiminto suoritetaan yhdelle kohteelle.

Taulukko 4.1. Ajastettujen toimintojen värikoodit ja tiheys

Kaikki ajastetut toiminnot pyritään suorittamaan yöaikaan tai viikonloppuisin, jotta hallinta ja monitorointi ei kuormita verkkoa toimistoaikana. Ainoastaan RO:n raporttitietokantaa päivitetään toimistotunteina arkipäivinä. Se tapahtuu kolmen tunnin välein. Päivitykset alkavat aina keskiyöllä ja yhdelle päivityskerralle on varattu aikaa 30 minuuttia. Agenttien jakelujärjestelmä asentaa ja testaa agenttien toimintaa kerran päivässä jokaiselle palvelimelle. Tähän toimintoon on varattu vuorokaudesta kuusi tuntia. Tämä kuuden tunnin kiintiö suoritetaan neljässä 1,5 tunnin jaksossa alkaen klo 0.30, 3.30, 18.30 ja 21.30. Jakelujärjestelmä on ajastettu siten että viiden minuutin välein agentit tarkistetaan neljästä laboratorioympäristöstä (palvelinklusterista).

Palvelimien, kytkimien ja levyjärjestelmien yksityiskohtaiset tiedot (Get details) kerätään kerran päivässä. ”Get details” kerää tietoja mm. vyökykkeistä, LUN:eista, järjestelmän sovelluksien versioista ja tallennusverkon topologiatiedot. Palvelimien tietojen keräämiseen on varattu vuorokaudesta viisi tuntia alkaen klo 2.00, 5.00, 7.00, 20.00 ja 23.00. Näinä tunteina tiedot kerätään yhteensä 200 palvelimesta, 50 palvelinta 15 minuutin välein. Lisäksi 6.30-7.00 välinen aika on varattu kytkimien ja levyjärjestelmien tietojen keräämiseen.

Taulukko 4.2. Ajastettujen toimintojen ajoittuminen vuorokauden eri hetkillä (ma-pe)

	0	5	10	15	20	25	30	35	40	45	50	55
0	RO Päivitys						SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
1	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
2	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		
3	RO Päivitys						SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
5	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		
6	RO Päivitys						SE DD kytkimet ja levyjärjestelmät					
7	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		
8												
9	RO Päivitys											
10												
11												
12	RO Päivitys											
13												
14												
15	RO Päivitys											
16												
17												
18	RO Päivitys						SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
19	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
20	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		
21	RO Päivitys						SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
22	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
23	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		

Myös SIM kerää tietoja hallittavista palvelimista. Koska SIMin toimintoja käytetään ainoastaan agenttien jakelujärjestelmään, sen tietojen kerääminen voidaan rajoittaa yhteen kertaan viikossa. Arkipäivien yöt on varattu jo muille ajastetuille toimenpiteille, joten SIMin tietojen keräys suoritetaan lauantaisin ja sunnuntaisin klo 11.00-17.00 pois lukien RO:n raporttitietokannan päivitykset klo 12.00 ja 15.00. Tänä aikana SIM kerää

tiedot kymmenestä laboratorioympäristöstä aina puolen tunnin välein. Yhtenä päivänä voidaan siis kerätä tiedot sadasta ympäristöstä ja viikonlopun aikana 200 ympäristöstä.

Taulukko 4.3. Ajastettujen toimintojen ajoittuminen vuorokauden eri hetkillä (la-su)

	0	5	10	15	20	25	30	35	40	45	50	55
0	RO Päivitys						SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
1	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
2	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		
3	RO Päivitys						SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
5	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		
6	RO Päivitys						SE DD kytkimet ja levyjärjestelmät					
7	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		
8												
9	RO Päivitys											
10												
11	SIM D x 10 ympäristöä						SIM D x 10 ympäristöä					
12	RO Päivitys						SIM D x 10 ympäristöä					
13	SIM D x 10 ympäristöä						SIM D x 10 ympäristöä					
14	SIM D x 10 ympäristöä						SIM D x 10 ympäristöä					
15	RO Päivitys						SIM D x 10 ympäristöä					
16	SIM D x 10 ympäristöä						SIM D x 10 ympäristöä					
17												
18	RO Päivitys						SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
19	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
20	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		
21	RO Päivitys						SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
22	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4	SE A x 4
23	SE DD x 50			SE DD x 50			SE DD x 50			SE DD x 50		

Näiden ajastettujen toimintojen lisäksi SE kerää tietoa palvelimien, kytkimien ja levyjärjestelmien suorituskyvystä tunnin välein. Palvelimista suorituskykytietoa kerätään mm. prosessoreista, muisteista, paikallisista kiintolevyistä ja kuituporteista. Kytkimistä tietoa kerätään myös porttien suorituskyvystä. Levyjärjestelmistä tietoa kerätään porttien, kontrollerien ja kiintolevyjen suorituskyvystä.

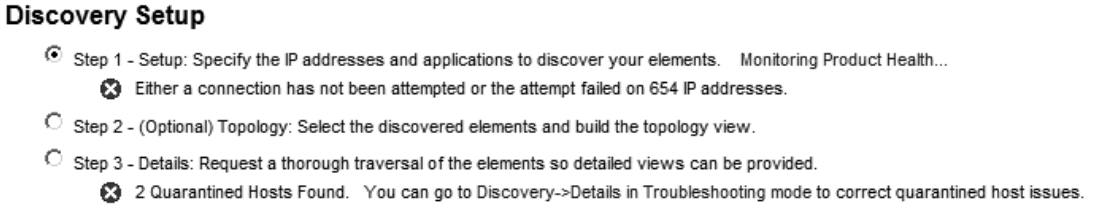
4.1.2 HP Storage Essentials SRM - Asetukset

Hallittavat laitteet on lisättävä sekä SE:hen ja SIMiin, jotta ne tietävät, mitä laitteita niiden tulee hallita. Lisäksi edellä mainitut ajastetut toiminnot ja automaattiset tapahtumiin reagoinnit (politiikat) pitää määrittää sovelluksiin. Tässä kappaleessa kerrotaan SE:hen tehtävät konfiguroinnit, jotta sen avulla voidaan hallita ja monitoroida tallennusverkkoa.

Laitteiden lisäys ja tietojen kerääminen

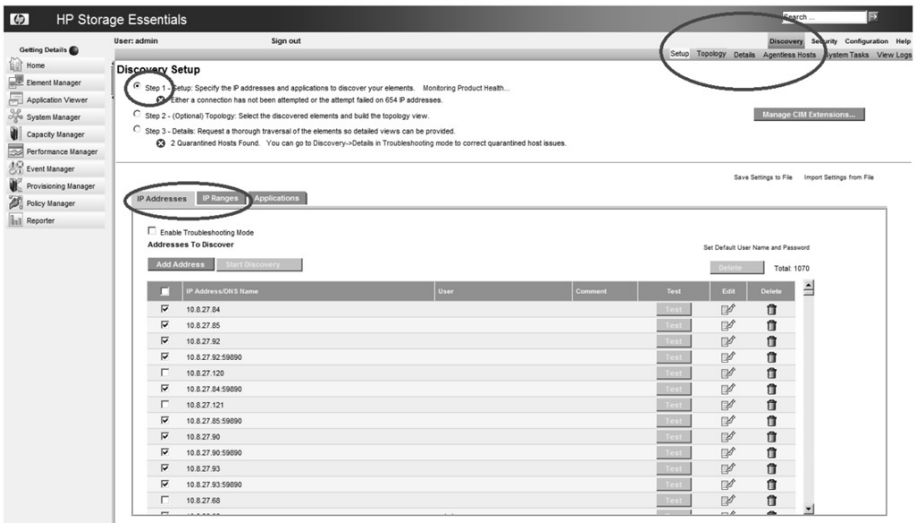
SE:n hallittavien laitteiden lisäys tapahtuu kolmessa vaiheessa (kuva 4.2). Ensimmäisessä vaiheessa (Step 1) SE:lle määritellään laitteiden IP-osoitteet tai DNS-nimet. IP-osoitteet voidaan lisätä myös IP-aliverkkona tai osoitealueena. Tässä vaiheessa SE tunnistaa laitteen tyypin ja sen käyttämät hallintaprotokollat. Mikäli hallittava laite käyttää jotain muuta porttia kuin protokollan oletusporttia, voidaan portti

määritellä IP-osoitteen tai DNS-nimen perään kaksoispisteen avulla (esimerkiksi 10.9.241.59:59890).



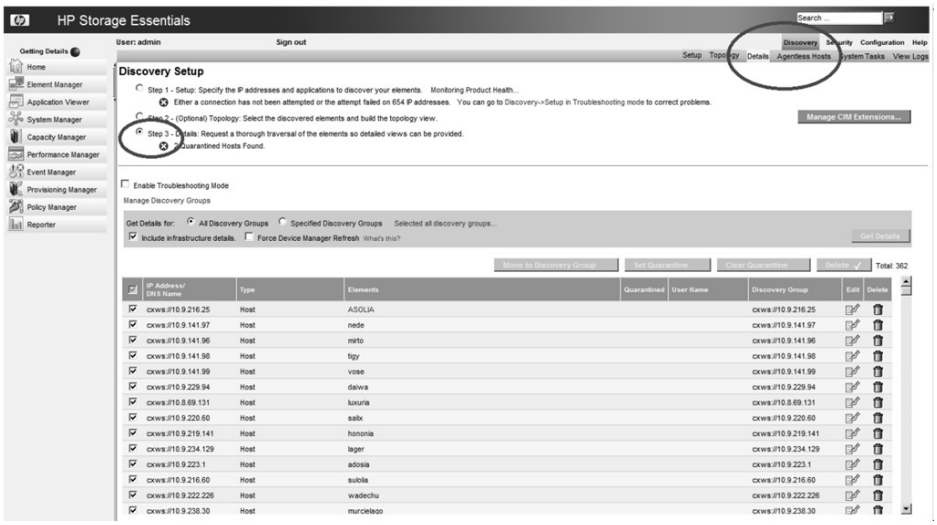
Kuva 4.2. SE:n kolme eri vaihetta laitteen tietojen keräämiseksi

Laitteiden lisäämiseen päästään SE:n näkymän oikeasta yläkulmasta *Discovery → Setup* (kuva 4.3). Yksittäinen laite voidaan lisätä klikkaamalla *Add Address*. Usean laitteen lisääminen kerrallaan tapahtuu välilehdeltä *IP Ranges.*



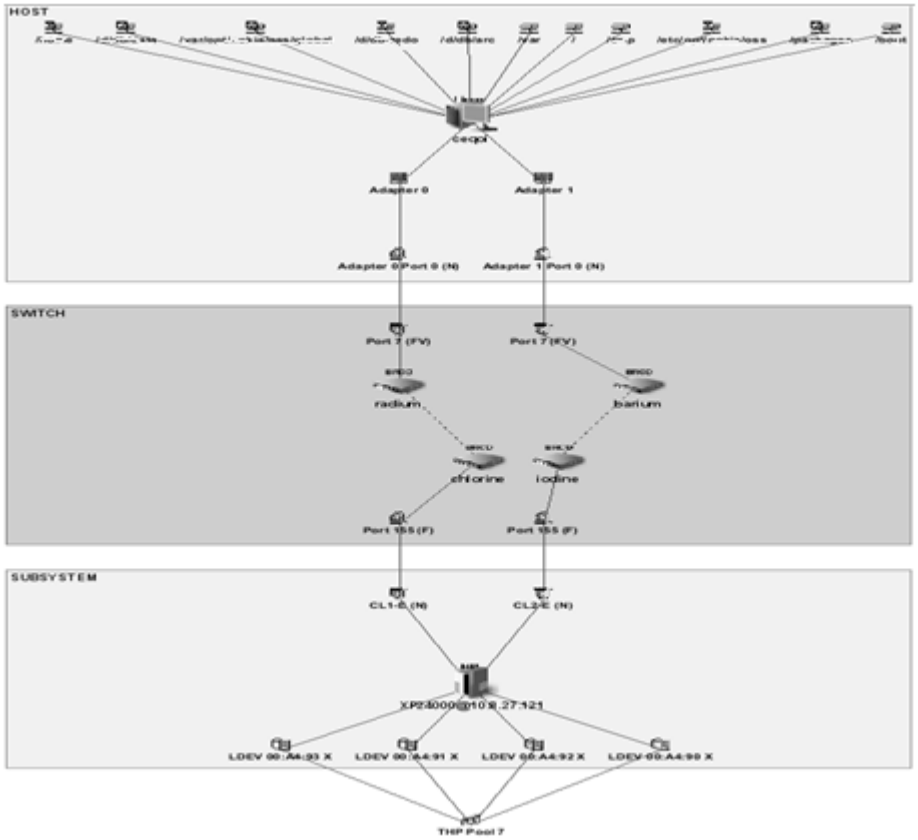
Kuva 4.3. Laitteiden lisäämisen ensimmäinen vaihe

Vaiheessa kaksi (Step 2) voidaan määritellä hallittavien laitteiden välinen topologia. Tämä vaihe suoritetaan myös vaiheessa kolme (Step 3), joten sitä ei ole pakko tehdä erikseen. Vaiheen kaksi avulla voidaan määritellä topologia nopeasti ilman, että laitteista kerätään muuta tietoa. Vaihe kolme kerää yksityiskohtaiset tiedot (Get Details) hallittavista laitteista ja niiden välisistä riippuvuuksista (kuva 4.4). Vaiheen kolme laitelistaan ilmestyvät ainoastaan laitteet, joihin SE on saanut hallintayhteyden jollakin tuetulla protokollalla. Esimerkiksi vaiheessa yksi lisätty palvelin, jolle ei ole asennettu agenttia, ei ilmesty tähän listaan. Jos agentti häviää palvelimelta, esimerkiksi käyttöjärjestelmän uudelleenasennuksen yhteydessä, SE asettaa kyseisen palvelimen karanteeniin. Tällaisesta palvelimesta ei kerätä tietoa ennen kuin karanteeni poistetaan valitsemalla kyseinen laite ja painamalla *Clear Quarantine*-painiketta.



Kuva 4.4. Laitteiden lisääminen kolmas vaihe (Get details)

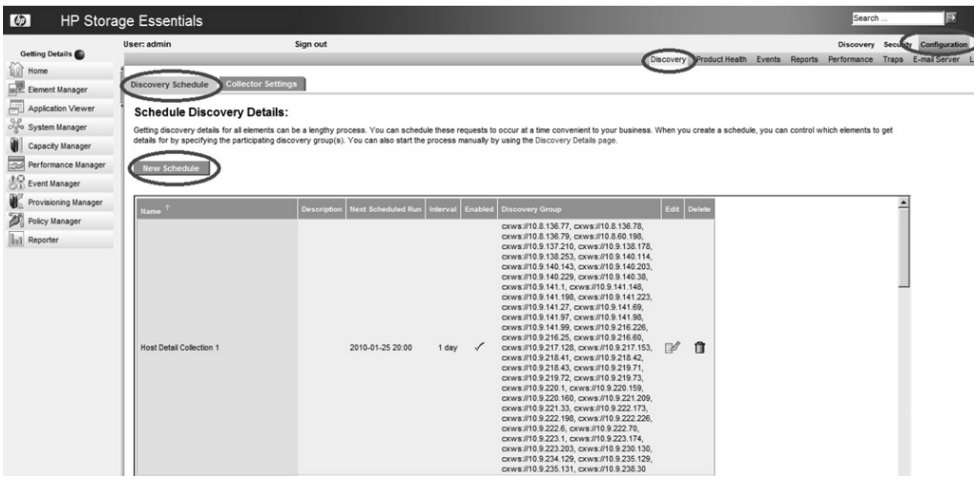
Jos laitteen lisääminen ja tietojen kerääminen on onnistunut, laitteen tietoja voidaan tarkastella esimerkiksi etsimällä se *Element Managerin* kautta. Kuvassa 4.5 on erään palvelimen topologia. Se näyttää koko ketjun palvelimen loogisista osioista aina levyjärjestelmän fyysisiin levyalueisiin asti. Topologia on jaettu tallennusverkon kolmeen osa-alueeseen (palvelin-, kudos- ja tallennuskerros).



Kuva 4.5. Palvelimelle levyjärjestelmästä näytettyjen levyjen siirtoketju tallennusverkon läpi

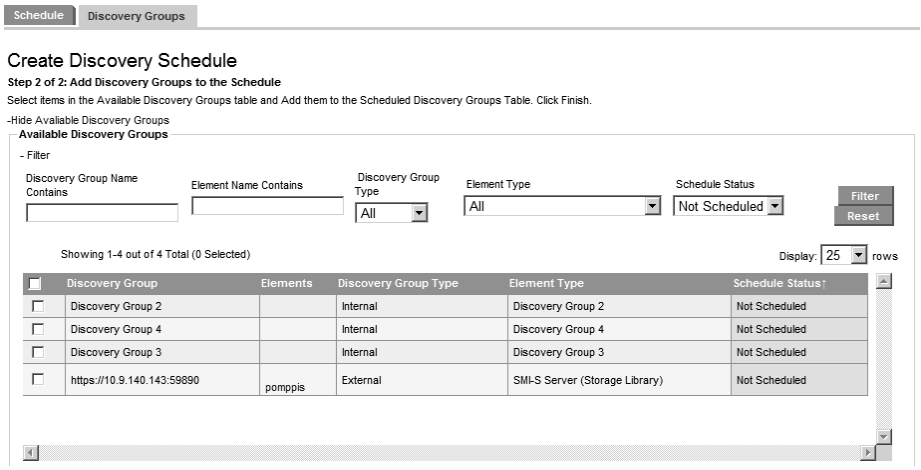
Ajastetut toiminnot

Jotta SE:n tiedot pysyvät ajan tasalla täytyy sen kerätä tietoja säännöllisin väliajoin hallittavista laitteista. Yksityiskohtaisten tietojen kerääminen (Details) voidaan ajastaa valitsemalla SE:n näkymän oikeasta yläkulmasta *Configuration* → *Discovery* → *Discovery Schedule* → *New Schedule* (kuva 4.6).



Kuva 4.6. Yksityiskohtaisten tietojen keräämisen ajastaminen SE:ssä

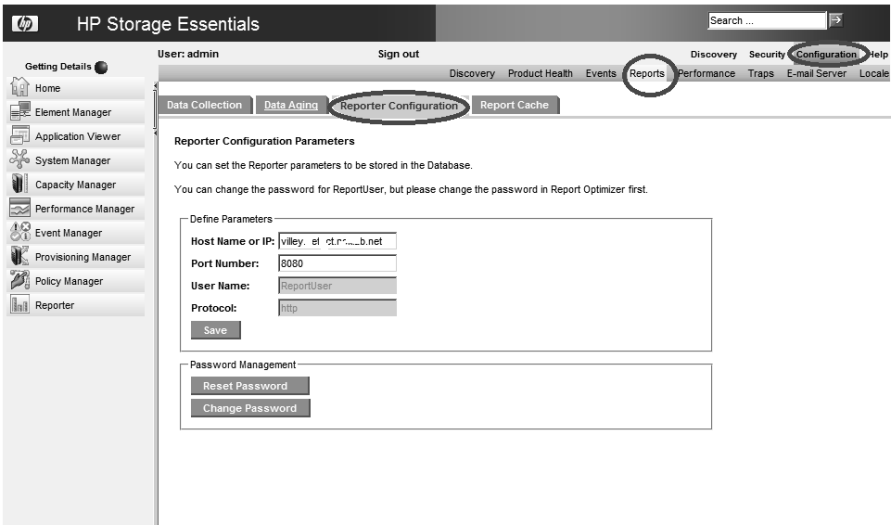
Ensimmäisellä välilehdellä määritellään ajastukselle nimi ja ajankohta. Toisella välilehdellä määritellään laitteet, jotka halutaan kyseiseen ajastukseen mukaan. Laitteet, joita ei ole vielä ajastettu, voidaan valita helposti käyttämällä suodatusta. Painamalla *Filter*+-näppäintä ja valitsemalla pudotusvalikosta *Not Scheduled*, saadaan listaan ainoastaan ajastamattomat laitteet (kuva 4.7).



Kuva 4.7. Uuden ajastetun toiminnon luominen SE:hen

RO:n raporttitietokanta määritellään valitsemalla *Configurations* → *Reports* → *Reporter Configurations* (kuva 4.8). Host name -kenttään kirjoitetaan palvelimen IP-osoite tai DNS-nimi, jossa RO sijaitsee. Portti-kentään kirjoitetaan RO:n käyttämä portti (oletuksena 8080). SE:n ja RO:n versiosta 6.2.0 lähtien oletustunnus RO:hon on *ReportUser*, jonka RO luo sovelluksen asennuksen tai päivityksen aikana.

Raporttitietokannan päivitystiheys määritellään välilehdeltä *Report Cache.* Kyseiseltä välilehdeltä voidaan myös pakottaa RO päivittämään raporttitietokanta heti.



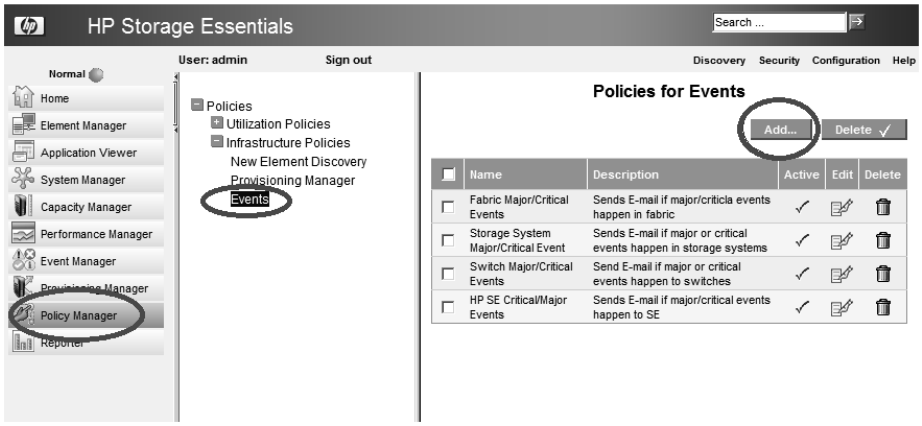
Kuva 4.8. Report Optimizer-sovelluksen lisääminen SE:hen

Laitteiden suorituskykyyn liittyvä tiedon keräys ajastetaan valitsemalla *Configuration → Performance*.

Politiikat

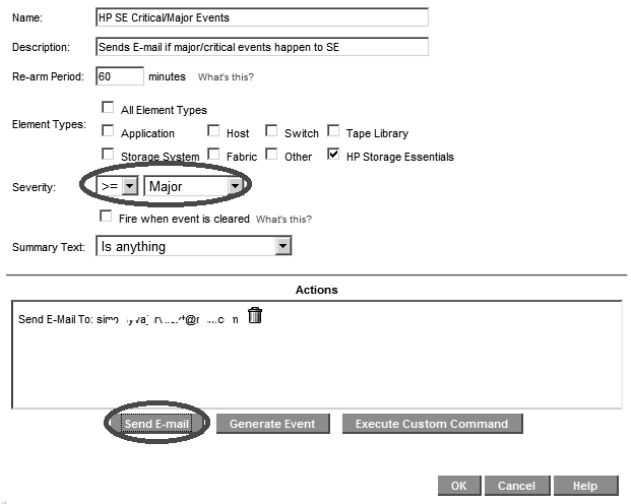
Policy Managerilla voidaan luoda sääntöjä, joiden avulla SE osaa reagoida automaattisesti tietyn tyyppisiin tapahtumiin. SE voidaan esimerkiksi konfiguroida lähettämään sähköpostia ylläpitohenkilöille tallennusverkossa tapahtuneista muutoksista. Vakavia muutoksia ovat mm. kytkimen osan hajoaminen tai kiintolevyn rikkoutuminen levyjärjestelmästä. Sähköpostin lähetystä varten SE:lle pitää määritellä sähköpostipalvelin. Postipalvelin voidaan määritellä valitsemalla *Configuration → E-mail Server.*

Policy Manageriin pääsee SE:n käyttöliittymän vasemmassa reunassa olevasta valikosta. Valitsemalla *Events* voidaan luoda sääntö tietynlaiseen tapahtumaan reagoimiseksi. Valitsemalla *Add…* lisätään uusi sääntö (kuva 4.9).



Kuva 4.9. SE:hen määritellyt politiikat, joilla reagoidaan automaattisesti tallennusverkossa tapahtuviin muutoksiin

Uusi sääntö saadaan luotua kirjoittamalla sille nimi, valitsemalla tapahtuman lähteen tyyppi (*Element Type*), kuinka vakavan tapahtuman pitää olla, että siihen reagoidaan *(Severity*) sekä lisäämällä ylläpitohenkilöiden sähköpostiosoitteet painamalla *Send E-mail*-näppäintä (kuva 4.10). SE on konfiguroitu lähettämään sähköpostia ylläpitäjille tilanteissa, joissa itse SE, levyjärjestelmä, kytkin tai muu kudoksen laite lähettää vakavan tai kriittisen tapahtumaviestin.

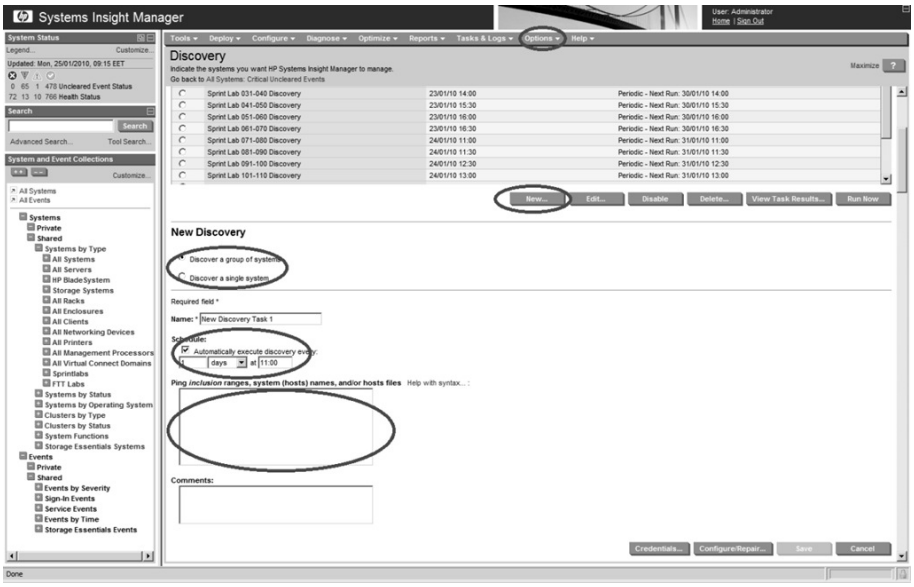


Kuva 4.10. Uuden politiikan määrittely

4.1.3 HP Systems Insight Manager - asetukset

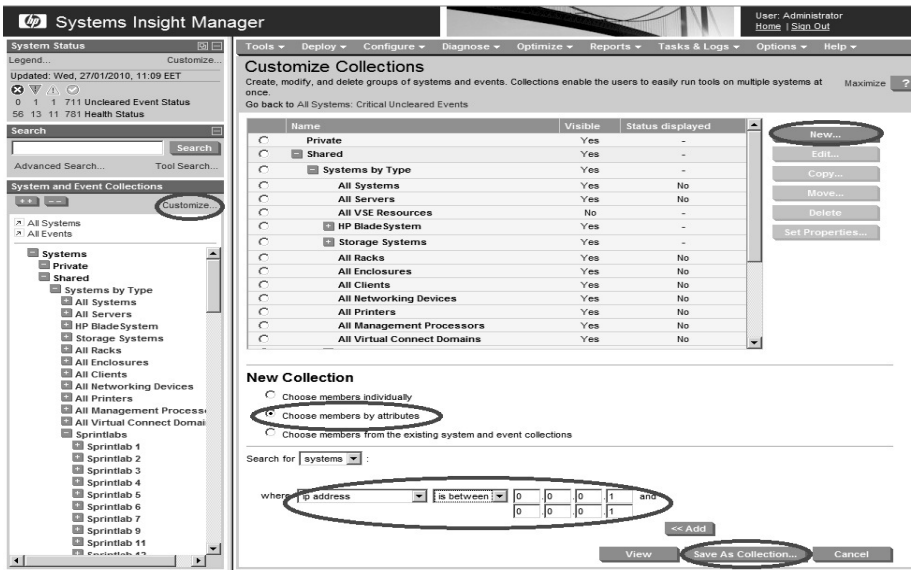
Hallittavat palvelimet lisätään SIM:iin valitsemalla käyttöliittymän yläosasta *pudostuvalikosta Options → Discovery → New.* Laitteita voi lisätä ryhmissä tai yksitellen valitsemalla joko *discover a group of systems* tai *discover a single system*. Lisääminen tapahtuu määrittelemällä laitteen IP-osoite tai usean laitteen IP-osoitealue,

DNS-nimi. Samalla tapahtuman voi ajastaa suoritettavaksi säännöllisin väliajoin (kuva 4.11). Ajastukset on selvitetty luvussa 4.1.1.



Kuva 4.11. Laitteiden lisääminen SIM:iin

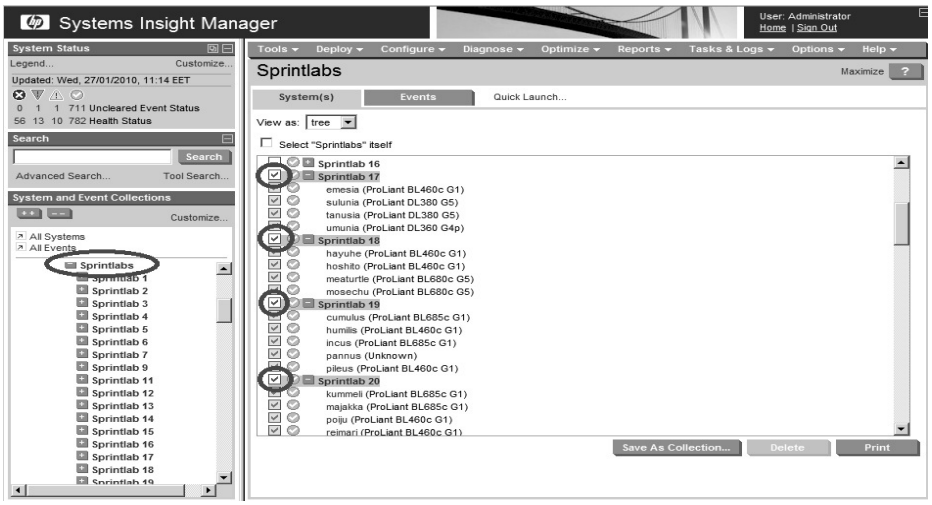
Kun laitteet on lisätty SIMiin, niistä kannattaa tehdä helpommin hallittavia kokonaisuuksia. Koska palvelimet kuuluvat aina johonkin laboratorioympäristöön (klusteriin), kannattaa saman ympäristön palvelimet lisätä yhteen kokoelmaan (collection). Kokolmia voi lisätä valitsemalla SIMin käyttöliittymän vasemmasta laidasta *Customize…* → *New…* → *Choose members by attributes* → *where ip address is between.* Kirjoittamalla ympäristön palvelimien IP-alueen niille varattuihin kenttiin ja tämän jälkeen painamalla Save *collection as…*, saadaan luoduksi kokoelma (kuva 4.12). Datakeskuksen kaikki laboratorioympäristöt on tallennettu omaksi kokoelmakseen tällä tavalla.



Kuva 4.12. Laitekokoelmien muodostaminen SIM:iin

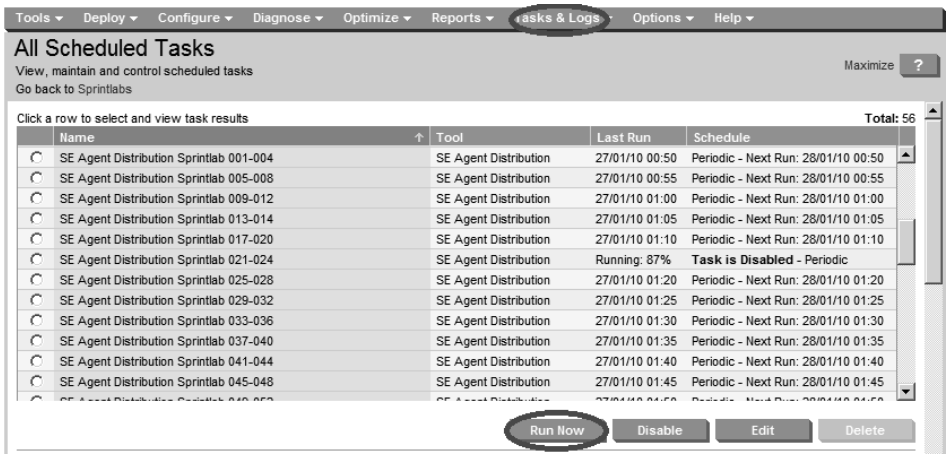
SIM:iin voidaan tehdä räätälöityjä työkaluja (Custom tools), joita voidaan ajaa esimerkiksi hallittavissa palvelimissa. Työkaluja voidaan luoda graafisen käyttöliittymän avulla valitsemalla SIM:n käyttöliittymän valikosta *Tools → Custom tools → New Custom tools… → Remote tool → Next* tai komentorivityökalun ja XML-tiedoston avulla. Agenttien jakelemiseksi on luotu kaksi työkalua, joista toinen asentaa tarvittaessa agentin kohdekoneille ja toinen pakottaa uudelleenasennuksen käyttämällä asennusskriptin force-lippua. Näiden työkalujen XML-tiedostot ovat liitteessä 1. Työkalu lisätään XML-tiedostosta SIM:iin komennolla *mxtool.exe –a –f <hakemistopolku ja tiedostonimi työkaluun>*. Ohjelma *mxtool.exe* sijaitsee hakemistossa *C:\Program Files (x86)\HP\Systems Insight Manager\bin*.

Agentteja jaetaan neljän laboratorioympäristön palvelimille kerrallaan. Ympäristöt saa valittua valitsemalle kuvan 4.13 mukaisesti. Työkalu ajastetaan suoritettavaksi valitsemalla *Tools → Custom Tools → SE Agent Distribution → Next → Schedule*. Ajastetulle tehtävälle on asetettava nimi ja vaihdettava valinta kohtaan *periodically*, jotta tehtävä suoritetaan säännöllisin väliajoin. Agentin jakelutyökalu on ajastettu suoritettavaksi kaikille laboratorioympäristöille. Ajastukset on selitetty luvussa 4.1.2. Jakelujärjestelmän toiminta selitetään tarkasti luvussa 4.2.



Kuva 4.13. Laitekokoelmien valinta ajastusta varten

Ajastettuja tehtäviä voi tarkastella, muokata ja suorittaa manuaalisesti valitsemalla SIM:n *valikosta Tasks & Logs → View All Scheduled Tasks*. Valitsemalla valikosta tietty tehtävä, voidaan myös tarkastella työkalun tuloksia. Tehtävän voi suorittaa manuaalisesti painamalla *Run Now*-näppäintä ja sitä voi muokata *Edit*-näppäimellä. Tehtävän ajastuksen voi pysäyttää painamalla *Disable* (kuva 4.14).



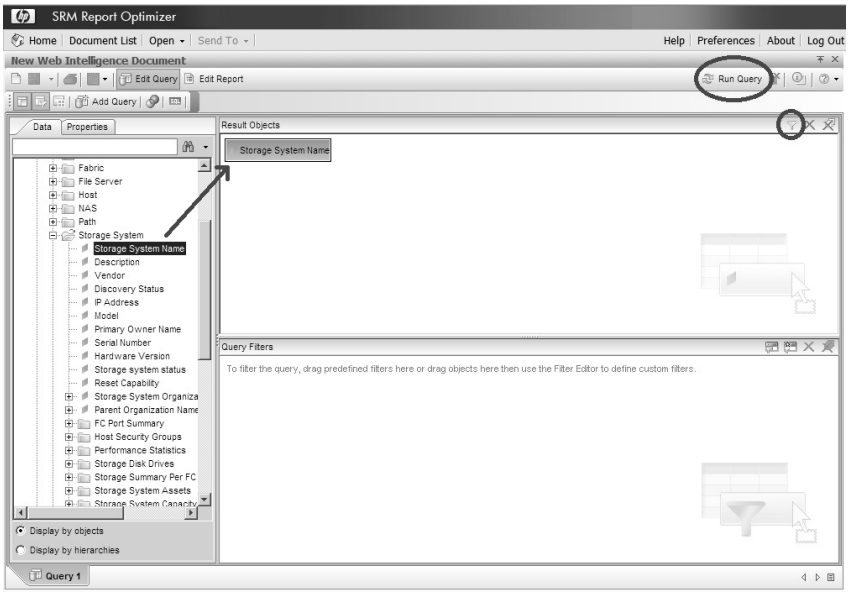
Kuva 4.14. Kaikki ajastetut toiminnot näkee *All Scheduled Tasks* --sivulta

4.1.4 Raportointi

BusinessObjects XI:hin perutuva Report Optimizer koostuu kahdesta käyttöliittymästä, joista toinen on hallintakonsoli ja toinen raporttityökalu. Hallintakonsoliin pääsee selaimella osoitteesta http://<palvelin>:8080/CmcApp/logon.faces ja raporttityökaluun osoitteesta http://<palvelin>:8080/InfoViewApp/logon.jsp.

Raporttien luonti

Versiosta 6.2.0 lähtien RO:hon on lisätty suuri määrä valmiita raportteja, jotka koostavat tietoa tallennusverkon eri osista. Valmiit raportit ovat hyvin yleiselle tasolla toteutettuja, joten tämän diplomityön tuloksena RO:hon on tehty myös joukko räätälöityjä raportteja, jotka vastaavat diplomityön kohteena olevan kehitettävän ympäristön tarpeita. BusinessObjects XI:llä voi tehdä usean tyyppisiä raportteja. RO:n kaikki raportit ovat tyypiltään *Web Intelligence Report.* Uusia raportteja voi tehdä valitsemalla raporttityökalusta *Document List* →*New* → *Web Intelligence Document* → *Report Connector*. Raportin teko aloitetaan valitsemalla haluttuja muuttujia näkymän vasemmasta laidasta olevasta valikosta ja siirtämällä muuttujia *Result Objects*-kenttään. Muuttujia voi suodattaa painamalla keltaista suppilo-kuvaketta *Result Objects*-kentän oikeassa yläkulmassa. Kun raportin tuloksia halutaan tarkastella, painetaan *Run Query*-näppäintä (kuva 4.15).

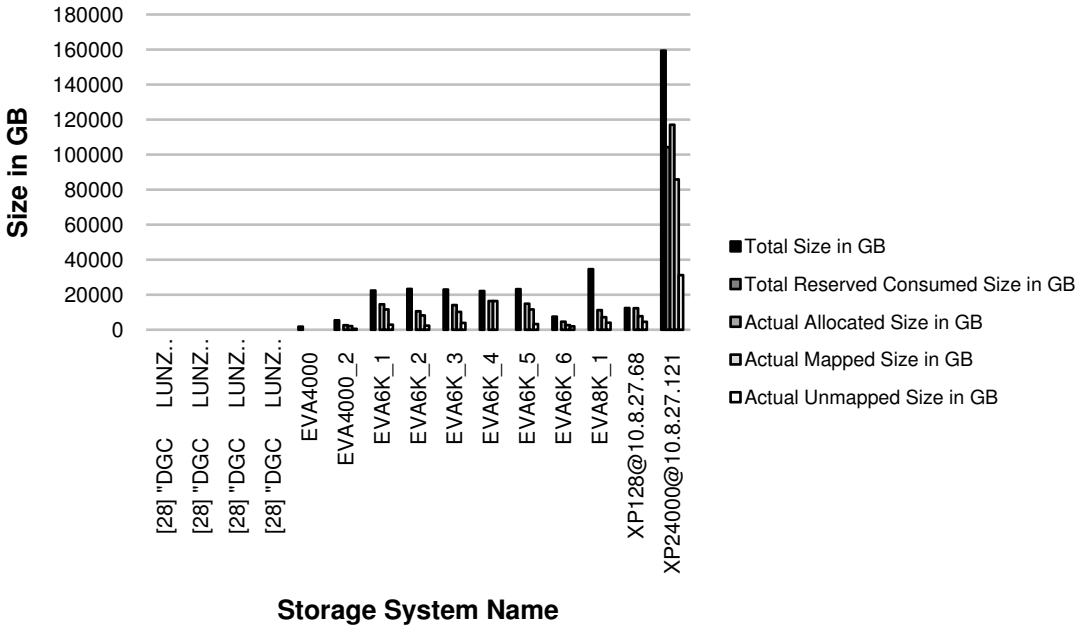


Kuva 4.15. Uuden raportin luominen RO:iin

Raportit

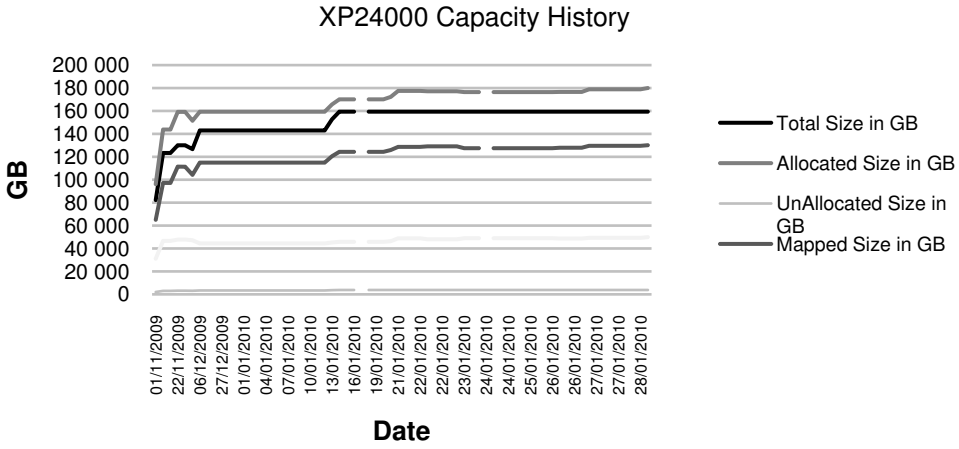
Räätälöidyt raportit on sijoitettu hakemistoon *Document List* → *Public Folders* → *Custom Reports*. Hakemisto sisältää kymmenen räätälöity raporttia, joiden nimet ja tarkoitukset on lueteltu alla.

- *Host Capacity Information*: Raportin avulla voidaan tarkastella yhden palvelimen tallennustilan muutosta ajan myötä viivadiagrammina.
- *SAN Switch Details*: Raportti kerää tallennusverkon kytkimien nimen, valmistajan, mallin, IP-osoitteen, laiteohjelmiston version, kudoksen nimen, toimialueen tunnisteen sekä käytettyjen porttien määrän ja porttien yhteismäärän ja listaa tulokset taulukkoon.
- *SAN Switch Port Details*: Raportti kerää tiedot kytkimien porttien tyypistä, nopeudesta, lähetys- ja vastaanottonopeuksista sekä virhemääristä. Raportti piirtää porttien nopeudet kuvaajaan. Tuloksia voidaan suodattaa kytkimen nimen, portin numeron tai portin tyypin mukaan.
- *Storage Capacity*: Raportti näyttää kaikkien levyjärjestelmien kapasiteettitiedot kuvaajassa. Jokaisesta levyjärjestelmästä kerätään tiedot fyysisen levytilan kokonaismäärästä, käytetystä fyysisestä levytilasta, allokoidusta levytilasta, allokoidusta levytilasta, joka näytetty palvelimille ja allokoidusta levytilasta, jota ei ole näytetty palvelimille (kuva 4.16).



Kuva 4.16. Tallennusverkon levyjärjestelmien kapasiteettitiedot

- *Storage Capacity History:* Raportti näyttää tietyn levyjärjestelmän kapasiteetin muutoksen ajan myötä (kuva 4.17).

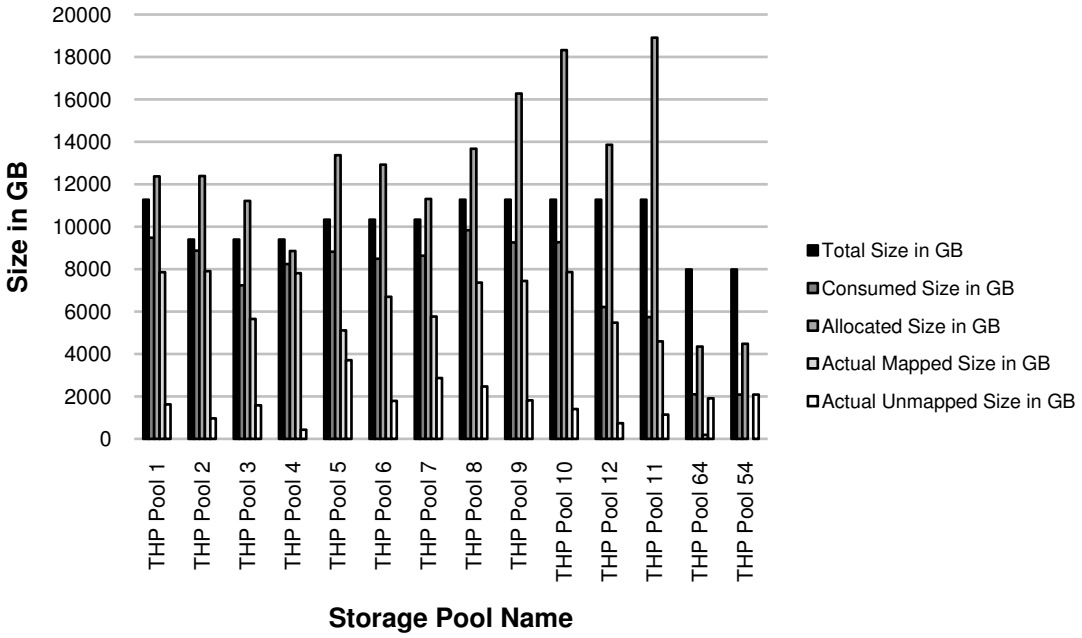


Kuva 4.17. XP24000-levyjärjestelmän kapasiteetin muuttuminen ajan myötä

- *Storage System Details:* Raportti kerää kaikista levyjärjestelmistä tietoa ja koostaa sen taulukkoon. Taulukossa on tietoa levyjärjestelmän valmistajasta, mallista, laitteistoversioista, tallennuskapasiteetista, porttimäärästä, kontrollerien mallista ja laitteistosovellusversiosta.
- *Switch Port Usage History:* Raportti näyttää tietyn kytkimen porttien yhteismäärän ja käytettyjen porttien määrän muutokset ajan myötä.
- *XP24000 Hosts, Host Group, LUN, LDEV, Volume Size:* Raportti kerää kahteen eri taulukkoon tietoa XP24000 levyjärjestelmään kytketyistä palvelimista.

Ensimmäisessä taulukossa on lueteltuna mihin XP24000:n porttiin ja Host Security Groupiin tietty palvelin on kytketty. Toinen taulukko listaa palvelimiin liitettyjen LUN:ien, levyjärjestelmän levyn nimen (LDEV) ja loogisen levyn koon.

- *XP24000 Port Infomation:* Raportti koostaa kolmeen taulukkoon tietoa XP24000 levyjärjestelmän porteista. Ensimmäinen taulukko sisältää portteihin liitetyt Host Security Groupit ja laskee niiden määrän. Tuloksia voi suodattaa tietyn portin perusteella. Toinen taulukko näyttää portteihin liitetyt Business Copy (varmuuskopio) Host Security Groupit. Kolmas taulukko listaa portteihin liitettyjen palvelimien määrän.
- *XP24000 THP Pool Allocation:* Raportti näyttää kuvaajassa XP24000-levyjärjestelmän Thin Provisioning Poolien kapasiteettitiedot. Jokaisesta poolista näytetään kokonaiskapasiteetti, allokoitu levytila, käytetty levytila ja käyttämätön levytila (kuva 4.18).



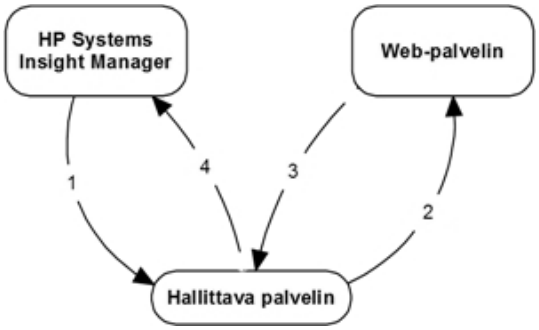
Kuva 4.18. XP24000 Thin Provisioning --levyalueiden kapasiteettitiedot

4.2 Agenttien automaattinen jakelujärjestelmä

Koska kehitettävässä ympäristössä olevat palvelimet toimivat ohjelmistokehitysprojektin alustana ja niiden käyttöjärjestelmä saatetaan asentaa uudelleen päivittäin tai viikoittain, eivät SE:n omat työkalut agenttien jakelemiseen palvelimille ole toimivia. Tästä syystä yksi tämän diplomityön tavoite oli kehittää agenttien jakelujärjestelmä, joka soveltuu tähän ympäristöön.

4.2.1 Arkkitehtuuri

Jakelujärjestelmä koostuu kolmesta komponentista, jotka ovat HP SIM, Web-palvelin ja hallittavat palvelimet (kuva 4.19). SIM toimii jakelujärjestelmän toimeenpanevana komponenttina. SIM:iin voidaan luoda räätälöityjä työkaluja (Custom tools), joita voidaan suorittaa halutuille palvelimille. Web-palvelimen tarkoitus on säilyttää agenttien asennustiedostoja ja jakaa niitä hallittaville palvelimille.



Kuva 4.19. SE:n agenttien automaattisen jakelujärjestelmän komponentit

Ensimmäisessä vaiheessa SIM kopioi asennusskriptin hallittavalle palvelimelle SSH:n avulla, antaa asennusskriptin tiedostoille suoritusoikeudet ja aloittaa skriptin suorittamisen. Mikäli agenttia ei ole asennettu hallittavalle palvelimelle tai se on asennettu puutteellisesti, hallittava palvelin ottaa yhteyden web-palvelimeen (toinen vaihe) ja lataa asennustiedostot HTTP-protokollan avulla (kolmas vaihe). Asennus joko onnistuu tai epäonnistuu, mistä hallittava palvelin raportoi SIM:lle. Asennustiedostot voitaisiin kopioida SIM:n avulla suoraan palvelimelle, samalla kun se kopioi asennnusskriptin tiedostot, mutta tämä aiheuttaisi turhaa kuormitusta verkkoon, koska asennustiedostojen yhteiskoko on noin 40 MB. On järkevämpää ladata asennustiedostot vain tarvittaessa.

Web-palvelimen hakemistossa, jossa asennustiedostot sijaitsevat, pitää olla tietynlainen hakemistorakenne, jotta asennusskripti osaa ladata oikeat tiedostot. Hakemistossa pitää olla alihakemistot ”rhel3”, ”rhel4” ja ”rhel5”, jotka kaikki sisältävät kansiot i386 ja x86_64. Nämä kansiot sisältävät asennustiedostot 32-bittiselle (i386) ja 64-bittiselle (x86_64) käyttöjärjestelmän versioille. Lisäksi juurihakemistossa pitää olla alihakemisto qlapi, jossa sijaitsee SNIA:n HBA-rajapinnan asennuspaketti. Juurihakemistossa sijaitsevat kaikille RHEL-versioille yhteinen asennustiedosto sekä agentin vaatima *compat-libstdc++-kirjasto*, joka asennetaan hallittavalle palvelimelle tarvittaessa. Alla on kuvattuna hakemistorakenne. Listauksessa hakemistot on lihavoitu ja tiedostot kursivoitu.

- **Juurihakemisto**
  - **rhel3**
    - **i386**
      - *APPQcime-Requires-<versio>-i386.rpm*

        - **x86_64**
            - *APPQcime-Requires-<versio>-x86_64.rpm*
    - **rhel4**
        - **i386**
            - *APPQcime-Requires-<versio>.i386.rpm*
        - **x86_64**
            - *APPQcime-Requires-<versio>-x86_64.rpm*
    - **rhel5**
        - **i386**
            - *APPQcime-Requires-<versio>-i386.rpm*
        - **x86_64**
            - *APPQcime-Requires-<versio>-x86_64.rpm*
    - **qlapi**
        - *qlapi.tgz*
    - *APPQcime-<versio>-i386.rpm*
    - *compat-libstdc++-296.i386.rpm*

4.2.2 Sisäänkirjautuminen hallittaviin palvelimiin

Jotta SIM saa etäyhteyden hallittavaan palvelimeen, täytyy SIM:n tiedossa olla ylikäyttäjän (root) käyttäjätunnus ja salasana. Koska kehitettävässä ympäristössä on useita satoja palvelimia, niiden kaikkien sisäänkirjautumistietojen syöttäminen SIM:iin ei ole mielekästä. Hallittavien palvelimien salasanat saattavat myös vaihtua asennuskertojen välillä, jolloin ne pitäisi aina vaihtaa myös SIM:iin.

Parempi vaihtoehto sisäänkirjautumiseen on käyttää SSH-avaimeen perustuvaa sisäänkirjautumista. Lisäämällä SIM-palvelimen SSH-avaimen julkinen osa hallittavien palvelimien *authorized_keys-tiedostoon* mahdollistaa SIM:n sisäänkirjautumisen SSH:n avulla ilman käyttäjätunnusta ja salasanaa. Kyseinen tiedosto sijaitsee halutun käyttäjän kotihakemistossa. Esimerkiksi root-käyttäjällä kyseisen tiedoston hakemistopolku on */root/.ssh/authorized_keys*. Mikäli SIM on asennettu oletushakemistoon, sen SSH-avaimen julkinen osa löytyy hakemistopolusta *C:\Program Files\HP\Systems Insight Manager\config\sshtools\.dftSshKey.pub*.

Kehitettävä ympäristö sisältää jo hallintapalvelimia, joiden SSH-avain on lisätty hallittavien palvelimien authrorized_keys-tiedostoon. Tällaista palvelinta voidaan käyttää apuna SIM:n avaimen levittämiseen hallittaville palvelimille. Levittämistä varten on toteutettu skripti *add-ssh-key.sh* (Liite 2). Skripti on ohjelmoitu Perl-ohjemointikielellä, koska se on asennettuna oletuksena hallittaville palvelimille ja se on tehokas ohjelmointikieli linux-komentorivin käyttöön. Skriptille tarvitsee asettaa kolme asetusta, jotka löytyvät kyseisen skriptin alusta. Taulukkoon *filenames* pitää lisätä yksi tai useampi tiedosto, johon SSH-avain lisätään. Taulukkoon *keys* tulee lisätä yksi tai useampi SSH-avain, jotka halutaan lisätä kohdepalvelimelle. Lisäksi asetuksista löytyy

muuttuja *create_file*, joka määrää, yrittääkö skripti luoda *filenames*-taulukossa olevaa tiedostoa, jos sitä ei ole olemassa. Jos muuttujan arvo on *yksi* (totuusarvo on tosi), skripti yrittää luoda tiedoston ja jos sen arvo on *nolla* (totuusarvo on epätosi), skripti ei yritä luoda tiedostoa.

Halutun tiedoston löydyttyä, skripti tarkistaa onko sillä lukuoikeudet kyseiseen tiedostoon. Jos sillä on lukuoikeudet, skripti tarkistaa, onko kyseiseen tiedostoon lisätty jo *keys*-taulukossa määriteltyjä avaimia. Skripti poistaa taulukosta avaimet, jotka on lisätty jo, jotta niitä ei lisätä seuraavassa vaiheessa toiseen kertaan. Tämän jälkeen skripti tarkistaa, onko sillä kirjoitusoikeudet kyseiseen tiedostoon. Jos skriptillä on kirjoitusoikeudet, se avaa tiedoston ja lisää avaimet tiedoston loppuun. Niissä tapauksissa, että määriteltyä tiedostoa ei löytynyt eikä niitä voinut lisätä tai tiedostoon ei ollut luku- tai kirjoitusoikeuksia, skriptin suoritus lopetetaan.

SSH-avainta jakavaa skriptiä tarvitaan lähinnä Storage Essentialin käyttöönoton alkuvaiheessa. SSH-avain on lisätty myös hallittavien palvelimien asennuspinoon, jotta se lisätään automaattisesti, kun palvelin asennetaan uudelleen aivan alusta. Joitakin hallittavia palvelimia palautetaan myös varmuuskopioista, ja jos SSH-avain ei ole ollut lisättynä varmuuskopion tekohetkellä, se häviää aina palautuksen yhteydessä. Tällaisessa tilanteessa tätä skriptiä tarvitaan pidempään.

4.2.3 Asennusskripti

Skripti, joka suorittaa varsinaisen agentin asennuksen, koostuu kahdesta tiedostosta, jotka ovat *se-agent-install.sh* (Liite 3) ja *SEAgent.pm* (Liite 4). Skripti on toteutettu Perl-ohjelmointikielelllä. Tiedosto *se-agent-install.sh* sisältää skriptin pääohjelman ja *SEAgent.pm* pakkauksen (Package), jossa on pääohjelman käyttämät funktiot.

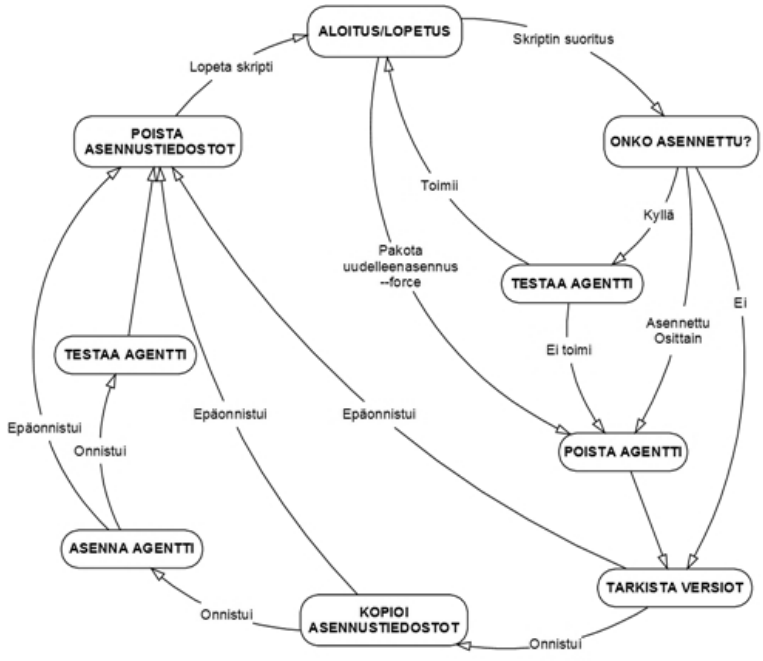
Alkuasetukset

Tiedoston *se-agent-install.sh* alussa on skriptin suorittamiseen tarvittavat alkuasetukset. Alkuasetus *server* on web-osoite, jossa asennustiedostot sijaitsevat. Asetus on annettava muodossa *”http://<IP-osoite tai DNS-nimi>/hakemistopolku”*. Alkuasetus *dlretry* on uudelleenyritysten määrä, joka suoritetaan, jos asennustiedostojen lataus epäonnistuu. Hallittavalla palvelimella käytettävä väliaikaiskansio määritellään muuttujalla *tmpdir*. Asennustiedostot ladataan ja puretaan tähän kansioon. SNIA HBA -sovellusrajapinnan (qlapi) asennustiedostot sisältävät myös ohjelman rajapinnan poistamiseksi. Tämä tiedosto siirretään *sniadir*-muuttujan osoittamaan kansioon, jotta rajapinta voidaan poistaa esimerkiksi uudelleenasennuksen yhteydessä. Alkuasetus *version* määrittelee asennettavan agentin version. Asetusta käytetään ladattavien asennustiedostojen nimissä. Agentilla on linux-käyttöjärjestelmässä yksi kirjastoriippuvuus. Se tarvitsee *compat-libstdc++*-kirjaston, jotta agentin voi asentaa. Asennusskripti osaa asentaa tämän kirjaston tarvittaessa. Muuttujaan *dependency* määritellään *yksi* (tosi), jos halutaan että skripti asentaa riippuvuuden. Asettamalla *nolla* (epätosi) riippuvuutta ei

asenneta. Jos kirjasto puuttuu hallittavalta palvelimelta ja asetus on *nolla*, agenttia ei asenneta palvelimelle.

Toimintalogiikka

Asennusskripti noudattaa kuvan 4.20 mukaista toimintalogiikkaa. Kun skripti on kopioitu hallittavalle palvelimelle, se tarkistaa ensimmäiseksi, onko agentti asennettu jo palvelimelle. Jos agentti on asennettu, skripti testaa toimiiko agentti. Agentin toimiessa ohjelman suoritus lopetetaan. Jos agentti on asennettu puutteellisesti (APPQcime tai SNIA HBA rajapinta puuttuu) tai testaus toteaa, että agentti ei toimi, skripti poistaa vanhat asennukset ja yrittää asentaa agentin uudelleen. Tässä tapauksessa tai tapauksessa, että agenttia ei ole asennettu, suoritus siirtyy versioiden tarkistamiseen, jossa skripti etsii käyttöjärjestelmän version (RHEL 3-5) ja tarkistaa onko se 32-bittinen vai 64-bittinen. Jos versioita ei pystytä päättelemään, skripti poistaa mahdollisesti sinne jääneet asennustiedostot ja lopettaa ohjelman suorituksen. Versioiden selvityksen onnistuessa skripti osaa tietojen perusteella ladata Web-palvelimelta oikeat asennustiedostot. Jos asennustiedostojen kopioiminen onnistuu, skripti siirtyy agentin asennukseen. Jos kopioiminen epäonnistuu, skripti poistaa mahdolliset asennustiedostot ja lopettaa ohjelman suorituksen. Mikäli agentti asennetaan onnistuneesti, skripti testaa agentin toimivuutta ja tulostaa testin tiedot. Agentin asennuksen epäonnistuessa ja testauksen loputtua, skripti poistaa asennustiedostot ja lopettaa ohjelman suorituksen.



Kuva 4.20. Asennuskriptin toimintalogiikka

SEAgent – pakkauksen funktiot

Tässä kappaleessa on kerrottu asennusskriptin käyttämien funktioiden tarkoitus ja toiminta. Funktiot noudattelevat kuvassa 4.20 esitetyn toimintalogiikan tiloja.

new: Perl ei varsinaisesti ole olio-ohjelmointikieli, mutta pakkauksista voidaan luoda oliota kiertotien kautta. Funktio *new* luo SEAgent – olion, jonka kautta muihin funktioihin voidaan viitata jatkossa. Funktiolla ei ole parametreja ja sen paluuarvo on olio itse.

checkIfInstalled: Funktio tarkistaa, onko agentti asennettu kyseiselle hallittavalle palvelimelle. Se tarkistaa, onko SNIA HBA – rajapinta ja APPQcime–ohjelma asennettu. Funktiolla ei ole parametreja. Sen paluuarvoja ovat *APPQcime*, *SNIA* tai *BOTH*. Funktio palauttaa *APPQcime*, jos ainoastaan agenttiohjelma on asennettu, *SNIA,* jos ainoastaan SNIA HBA – rajapinta on asennettu ja *BOTH*, jos molemmat on asennettu.

verifyVersions: Tämä funktio selvittää, mistä käyttöjärjestelmän versiosta on kyse ja onko se 32-bittinen vai 64-bittinen. Funktiolla ei ole parametreja. Sillä on kaksi paluuarvoa *os* ja *platform*, joihin tallentuu käyttöjärjestelmän versio ja bittimäärä.

copyInstallationFiles: Funktio lataa asennustiedostot web-palvelimelta. Sillä on kuusi parametria. *Server* on asennustiedostot sisältävän palvelin osoite, *os* ja *platform* ovat *verifyVersions*-funktion paluuarvoja ja *dlretry*, *tmpdir* sekä *version* ovat skriptin alkuasetuksia. Funktion paluuarvo on *yksi* (tosi), jos lataaminen onnistui ja *nolla* (epätosi), jos se epäonnistui.

installAgent: Funktio asentaa agentin edellisen funktion lataamista asennustiedostoista. Sen parametreja ovat *tmpdir, os, server, version, ja dependency*, jotka ovat samoja kuin edelle mainitut ja *dependency* on asennuskriptin alkuasetus. Funktion paluuarvo on *yksi* (tosi), jos asennus onnistui ja *nolla* (epätosi), jos se epäonnistui.

testAgent: Funktio testaa, onko agentti päällä ja toimiiko se. Jos agentti ei ole päällä, funktio yrittää käynnistää sen. Funktiolla ei ole parametreja. Funktion paluuarvo on *yksi* (tosi), jos agentti on päällä ja *nolla* (epätosi), jos agentin käynnistäminen epäonnistuu.

deleteInstallationFiles: Funktio poistaa asennustiedostot ja asennuksen aikana käytettävät väliaikaishakemistot. Se myös siirtää SNIA HBA – rajapinnan poisto-ohjelman haluttuun kansioon. Funktiolla on kaksi parametria *tmpdir* ja *sniadir*, jotka ovat asennusskriptin alkuasetuksia. Funktiolla ei ole paluuarvoja.

deleteAgent: Funktio poistaa agentin ja HBA – rajapinnan. Sen ainut parametri on *sniadir*, jonka avulla funktio löytää rajapinnan poisto-ohjelman. Funktion paluuarvo on *yksi* (tosi), jos poistaminen onnistui ja *nolla* (epätosi), jos se epäonnistui.

Esimerkkituloste

Asennuskriptin esimerkkiulostulo on kuvattuna liitteessä 5. Siinä on tulosteet tapauksesta, jossa asennuskripti on pakotettu asentamaan agentti uudelleen komentoriviparametrilla –force. Tuloste näkyy SIMissa tehtävän tuloksissa (*Task Results*).

Rivillä yksi on kerrottuna, että uudelleenasennus on pakotettu. Riveillä kolme ja neljä ohjelma poistaa SNIA HBA – rajapinnan. Riveillä viisi ja kuusi ohjelma on selvittänyt, että käyttöjärjestelmän versio on 64-bittinen RHEL4. Seuraavaksi skripti on kopioinut asennustiedostot Web-palvelimelta, joista se on tulostanut rivit 6-8. Riveillä 9-14 skripti asentaa SNIA HBA – rajapinnan. RHEL4 vaatii, että rajapinnan moduuli pitää ladata manuaalisesti. Tämän tulostus on riveillä 15 ja 16. Hallittavalla palvelimella ei ole asennettuna compat-libstcdc++-296 – kirjastoa. Skripti lataa asennustiedoston web-palvelimelta ja asentaa kirjaston riveillä 17-22. Riveillä 23-26 skripti asentaa itse agentin (APPQcime). Loput rivit käsittelevät agentin testausta. Testi löytää kaksi HBA:ta, joissa molemmissa on yksi kuituportti. Ensimmäisen HBA:n (nolla) kautta testi löytää neljä SCSI-kohdetta, joiden LUN-numerot ovat 100, 200, 300 ja 400.

5 JOHTOPÄÄTÖKSET

Datakeskuksen palvelinmäärän kasvaessa siirtyminen tallennusverkon käyttöön on erittäin suositeltavaa. Tallennetun tiedon hallinta monimutkaistuu huomattavasti, jos tallennustilana käytetään ainoastaan palvelimen omia kiintolevyjä. Datan turvallinen säilytys ja korkean saatavuusasteen varmistaminen on vaikea toteuttaa tällä tavalla. Tallennuskapasiteettia ei myöskään voida tehokkaasti jakaa palvelimien kesken ilman fyysisten laitteiden siirtoa palvelimien välillä. Tallennusverkon ja keskitetyn tallennuslaitteen avulla voidaan parantaa tiedon turvallista tallennusta, suorituskykyä, saatavuutta sekä tehostaa tallennustilan reilua jakamista palvelimien kesken.

Tallennusverkon suurin ongelma on aina ollut laitteiden korkea hinta, minkä takia ne ovat pysyneet lähinnä suurien yrityksien ja valtion laitosten tekniikkana. Fibre Channelilla toteutettu tallennusverkko on dominoinut pitkään tallennusverkkomarkkinoita, minkä takia laitteiden hinta ei ole pudonnut toivotulla tavalla. Teknisesti yhtä hyvien toteutusten puuttuminen on pitänyt Fibre Channelin markkinaosuuden yli 80 prosentissa (2009). Tilanne tullee kuitenkin muuttumaan lähivuosina, koska alalle on tullut monia uusia toteutustekniikoita, jotka pyrkivät vähentämään FC:n markkinaosuutta halvemmilla laitekustannuksilla ja vähentämällä datakeskusessa tarvittavien erilaisten verkkojen määrää. Näitä tekniikoita ovat lähinnä Ethernet- ja IP-protokollaan perustuvat ratkaisut, jotka pyrkivät yhdistämään normaalin lähiverkon, tallennusverkon ja hallintaan tarvittavan liikeenteen samaan verkkoinfrastruktuuriin. Tällä pyritään vähentämään eri verkkojen ylläpitoon vaadittavia kustannuksia. FC:n pahimpia kilpailijoita tullee olemaan 10 gigabitin Ethernetiin perustuva FCoE ja IP- ja SCSI-protokollan yhdistävä iSCSI.

Tallennusverkosta muodostuu nopeasti monimutkainen järjestelmä, joka koostuu useista eri komponenteista. Palvelimet, kudoskerroksen laitteet sekä tallennukseen käytettävät massamuistilaitteet täytyy pystyä liittämään toisiinsa tehokkaasti ja turvallisesti. Tallennusverkon hallintaan ja monitorointiin voidaan käyttää useita eri protokollia, joista suosituimmaksi nykyään on noussut avoin SMI-S-protokolla, joka on rakennettu WBEM- ja CIM-protokollan päälle. SMI-S on nimenomaan tallennusverkon eri laitteiden hallintaan tarkoitettu protokolla. Hallintaprotokollien avulla tallennusverkon hallinta voidaan keskittää yhdelle sovellukselle. Tämän diplomityön kohteena olleen datakeskuksen tallennusverkon hallintaan valittiin HP Storage Essentials SRM-hallintasovellus, joka pääasiallisesti käyttää SMI-S-protokollaa laitteiden hallintaan ja monitorointiin. Vaikka SMI-S on ollut jo jonkin aikaa markkinoilla, sen rajapintaa on toteutettu harvaan laitteeseen suoraan. Tällaisiin laitteisiin SE saa yhteyden esimerkiksi

SMI-S-välityspalvelimen kautta, joka kommunikoi SE:n kanssa SMI-S-protokollalla, ja hallittavien laitteiden kanssa jollakin muulla protokollalla tai sovellusrajapinnan kautta. Kehitetyssä ympäristössä SE tarvitsee EVA-levyjärjestelmille, kytkimille sekä nauhakirjastolle omat hallintasovellukset, jotka sisältävät SMI-S-rajapinnan näiden laitteiden hallintaan. Avoimella hallintaprotokollalla voidaan teoriassa hallita kaikkia laitteita valmistajasta riippumutta. Käytännössä asia ei kuitenkaan ole niin yksinkertainen, koska hallintasovellukset sekä hallittavat laitteet on testattu ainoastaan tietyillä sovellus- ja laitteiston ohjelmistoversioilla. Uusien versioiden käyttöönotto saattaa koitua epämieluisaksi, koska kaikkien päivitettävien sovellusten ja ohjelmistojen määrä saattaa kasvaa todella suureksi.

HP Storage Essentials SRM on erittäin monipuolinen sovellus, joka pyrkii yhdistämään tallennusverkon kaikkien osien hallinnan yhteen pisteeseen. SE pyrkii olemaan riippumaton valmistajasta ja se tukee erittäin montaa laite- ja ohjelmistovalmistajaa. Ongelmaksi kuitenkin muodostuu se, että hallittavat laitteet eivät tue esimerkiksi SMI-S-protokollaa suoraan. Jotta SE ja hallittava laite voivat kommunikoida, täytyy niiden väliin asentaa jokin tietoa välittävä sovellus, joka pystyy kommunikoimaan sekä SE:n että hallittavan laitteen välillä. Tämän diplomityön kohteena oleva ympäristö on varsin homogeeninen ja koostuu pääasiassa HP:n valmistamista laitteista. Useat HP:n valmistamat laitteetkin tarvitsevat tällaisen välitysohjelmiston, jotta ne voidaan integroida SE:hen.

SE:n kyky kommunikoida monien laitteiden kanssa tekee siitä erittäin hyvän ohjelman tiedon keräämiseen tallennusverkosta ja sen komponenteista. Tieto on jäsennetty selkeästi laitekohtaisesti ja hallittavia laitteita voidaan hakea helposti nimen tai tyypin mukaan. Hallittavien komponenttien konfigurointi ei ole toteutettu niin hyvin kuin kerätyn tiedon selaaminen. SE sisältää Provisioning Manager-työkalun, jonka tarkoitus on saada konfiguroitua koko polku palvelimelta allokoituun levytilaan yhdestä näkymästä. Valitettavasti tämä ja moni muu SE:n työkalu on toteutettu Java-appletien avulla, jotka toimivat erittäin hitaasti. Pitkään käytössä olleessa tallennusverkossa on syntynyt tietynlaisia toimintatapoja ja nimeämiskäytäntöjä, joiden käytön jatkaminen SE:ssä on melko hankalaa.

SE:n eräs huono puoli on myös porttiperusteinen lisensointi. Tallennusverkon kasvaessa siinä olevien porttien määrä saattaa lisääntyä nopeasti uusien kytkimien tai palvelimien myötä. Tämä johtaa siihen, että on ostettava kalliimpi lisenssi SE:n käyttöön. Mikäli SE:llä halutaan hallita myös virtuaalikoneita, luetaan jokainen virtuaalikone myös yhdeksi portiksi. Ominaisuuden saa pois päältä, mutta tällöin virtuaalikoneista ei kerätä tietoa eivätkä ne näy SE:ssä.

LÄHTEET

[1] Troppens U., Müller-Friedt W., Wolafka R., Erkens R., Haustein N. Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, iSCSI, InfiniBand and FCoE, Second Edition. John Wiley & Sons Ltd. 2009. 564 p.

[2] Clark T. Designing Storage Area Networks: A Practical Reference for Implementing Fibre Channel and IP SANs, Second Edition. Addison Wesley. 2003. 592 p.

[3] Nikitan A., Poelker C. Storage Area Networks for Dummies, 2nd Edition. Wiley Publishing, Inc, Indiana. 2009. 438 p.

[4] Long J. Storage Networking Protocol Fundamentals. Cisco Press. 2006. 552 p.

[5] Judd J., Kruger D. Principles of SAN Design. Infinity Publishing. 2005. 469 p.

[6] Judd J. Multiprotocol Routing for SANs. 2nd Edition. Infinity Publishing. 2006. 354 p.

[7] ANSI X3.272-199x. Fibre Channel Arbitrated Loop (FC-AL) Rev. 4.5. New 1995, York American National Standards Institute.

[8] T11/Project 1674-D/Rev 7.7. Fibre Channel Switched Fabric – 4 (FC-SW-4) Rev 7.7. New York 2005, American National Standards Institute.

[9] Haikala I., Järvinen H-M. Käyttöjärjestelmät. Helsinki. Talentum Media Oy. 2004. 246 p.

[10] Lucchese F., Moore R., Tate J. Introduction to Storage Area Networks. 4th Edition. IBM Corp. 2006. 325 p.

[11] Meggyesi Z. Fibre Channel Overview [WWW]. [viitattu 23.12.2009] Saatavissa: http://hsi.web.cern.ch/HSI/fcs/spec/overview.htm.

[12] Storage Management Technical Spesification, Part 1 Common Architecture. Version 1.4.0, Revision 4. 2009. Storage Networking Industry Association.

[13] Secure Shell [WWW]. [viitattu 28.1.2010] Saatavissa: http://en.wikipedia.org/wiki/Secure_Shell

[14] Larochelle D., Rasasco N. How and Why More Secure Technologies Succeed in Legacy Markets: Lessons from the Success of SSH [WWW]. [viitattu 28.1.2010]. Saatavissa: http://www.cs.virginia.edu/~drl7x/sshVsTelnetWeb3.pdf

[15] Rajagopal M., Rodriguez E., Weber R. Fibre Channel Over TCP/IP (FCIP). RFC3821. IETF 2004. 74 p.

[16] Monia C., et al. iFCP – A Protocol for internet Fibre Channel Protocol Storage Networking. RFC4172. IETF 2005. 111 p.

[17] Satran J., et al. Internet Small Computer Systems Interface (iSCSI). RFC3720. IETF 2004. 257 p.

[18] T11/Project 1871-D/Rev 2.00. Fibre Channel Backbone – 5 (FC-BB-5) Rev. 2.00. New York 2009, American National Standards Institute.

[19] Tate J. Introduction to Fibre Channel over Ethernet, and Fibre Channel over Convergence Enhanced Ethernet. 2009 IBM Corp. 18 p.

[20] Pentakalos O. An Introduction to the InifiniBand Architecture [WWW]. [Viitattu 26.12.2009]. Saatavissa: http://www.oreillynet.com/pub/a/network/2002/02/04/windows.html

[21] HP Storage Essentials Storage Resource Management Enterprise Edition Software - Overview & Features [WWW]. [Viitattu 2.1.2010]. Saatavissa: http://h18000.www1.hp.com/products/storage/software/e-suite/se-index.html

[22] HP Storage Essentials v6.2: User Guide for the Standalone Configuration. 2009, Hewlett-Packard. 751 p.

[23] HP Storage Essentials v6.2: Installation Guide for the Standalone Configuration. 2009, Hewlett-Packard. 420 p.

[24] Conway S., Joseph E., Wu J., Walsh R. Extreme Computing: HP's New Blades Target HPC, Cloud Computing, and the Next-Generation Datacenter [WWW]. [Viitattu 5.1.2010]. Saatavissa: http://h20311.www2.hp.com/HPC/downloads/HP%27s%20new%20blades%20white%20paper.pdf

[25] HP Part Number: 356920-402. HP Systems Insight Manager 5.3 Technical Reference Guide. 2009, Hewlett-Packard. 758 p.

[26] Brocade SMI Agent Download [WWW]. [Viitattu 6.10.2010] Saatavissa: http://www.brocade.com/services-support/drivers-downloads/smi-agent/index.page

[27] 53-1001535-01. Brocade SMI Agent User's Guide (120.10.0). 2009, Brocade Communications Systems, Inc. 70 p.

[28] Part number: T5494-96303. HP StorageWorks CommandView EVA 9.1 user guide. 2009, Hewlett-Packard. 114 p.

[29] Partnumber: 344841-014. HP StorageWorks Interface Manager and Command View TL Version 2.3.01 User Guide. 2008, Hewlett-Packard. 210 p.

[30] HP Storage Essentials Storage Resource Management Report Optimizer 6.2 Installation Guide. 2009, Hewlett-Packard. 78 p.

[31] Part number: 5697-8037. HP StorageWorks 4x00/6x00/8x00 Enterprise Virtual Array user guide. 2009, Hewlett-Packard. 232 p.

[32] Part number: AE131-96072. HP StorageWorks XP24000/20000 Disk Array Owner's Guide. 2009, Hewlett-Packard. 66 p.

[33] Part number: 5697–7814. HP StorageWorks DC and DC04 SAN Backbone Director Switches, hardware reference guide. 2009, Hewlett-Packard. 256 p.

[34] 53-1001336-01. Fabric OS Administrator's Guide (v6.3.0). 2009, Brocade Communications Systems, Inc. 550 p.

[35] TC080301TB. HP BladeSystem c-Class architecture, technology brief, 2nd edition. 2008, Hewlett-Packard. 27 p.

[36] HP Proliant BL490c Generation 6 (G6) Server Blade [WWW]. [Viitattu 15.1.2010]. Saatavissa: http://h18004.www1.hp.com/products/quickspecs/13236_na/13236_na.html

LIITE 1: HP SIM - CUSTOM TOOL XML-TIEDOSTOT

```
1   <?xml version="1.0" encoding="windows-1252"?>
2   <tool-list>
3       <ssa-command-tool name="SE Agent Distribution"
    guid="00000017a477182c00000005000000f7" revision="1.0">
4           <category>Custom Tools</category>
5           <execute-as-user>
6               <root-admin />
7           </execute-as-user>
8           <toolbox-enabled value="true" />
9           <ssa-block>
10              <command command-type="stdout" log="false">cd /tmp/;
    chmod 700 se-agent-install.sh SEAgent.pm; ./se-agent-
    install.sh</command>
11              <copy-block>
12                  <source>E:\SE_Agent\se-agent-install.sh</source>
13                  <destination>/tmp/se-agent-install.sh</destination>
14                  <source>E:\SE_Agent\SEAgent.pm</source>
15                  <destination>/tmp/SEAgent.pm</destination>
16              </copy-block>
17          </ssa-block>
18          <attribute name="menu-sort-key">600</attribute>
19          <attribute name="menu-path">Tools|Custom Tools</attribute>
20          <attribute name="ui-editable">true</attribute>
21      </ssa-command-tool>
22  </tool-list>
```

```
1   <?xml version="1.0" encoding="windows-1252"?>
2   <tool-list>
3       <ssa-command-tool name="SE Agent Distribution (FORCE)"
    guid="00000017a477182c00000005000000f8" revision="1.0">
4           <category>Custom Tools</category>
5           <execute-as-user>
6               <root-admin />
7           </execute-as-user>
8           <toolbox-enabled value="true" />
9           <ssa-block>
10              <command command-type="stdout" log="false">cd /tmp/;
    chmod 700 se-agent-install.sh SEAgent.pm; ./se-agent-install.sh --
    force</command>
11              <copy-block>
12                  <source>E:\SE_Agent\se-agent-install.sh</source>
13                  <destination>/tmp/se-agent-install.sh</destination>
14                  <source>E:\SE_Agent\SEAgent.pm</source>
15                  <destination>/tmp/SEAgent.pm</destination>
16              </copy-block>
17          </ssa-block>
18          <attribute name="menu-sort-key">600</attribute>
19          <attribute name="menu-
    path">@MENU_PATH_TOOLS_CUSTOM</attribute>
20          <attribute name="ui-editable">true</attribute>
21      </ssa-command-tool>
22  </tool-list>
23
```

LIITE 2: ADD-SSH-KEY.SH

```
1   #!/usr/bin/perl -w
2   use warnings;
3   use strict;
4   # SETTINGS ============================================================
5   # filenames:  The possible filenames for the ssh key file, first
    found will be used
6   # keys:                       The SSH keys that will be added to the
    found file, keys that all already added are skipped
7   # create_file: Create new file if it does not exist, 1 creat, 0 don't
    create
8   my @filenames = (
9                           "/root/.ssh/authorized_keys2",
10                          "/root/.ssh/authorized_keys",
11                          #"<FILE 3>"
12                  );
13  my @keys =      (
14                          "<sshkeyhere>",
15                          #"<KEY 2>",
16                          #"<KEy 3>",
17                  );
18
19  my $create_file = 1;
20  #
    ======================================================================
21
22  my $filename = "";
23
24  # Check if the files exists
25  for( @filenames ) {
26          # Select first file that exists and exit loop
27          if( -e $_ ) {
28                  $filename = $_;
29                  last;
30          }
31          # Create file if it does not exist, and exit loop
32          elsif( $create_file ) {
33                  print "Create new file '$_'.\n";
34                  qx(touch $_);
35                  qx(chmod 644 $_);
36                  $filename = $_;
37                  last;
38          }
39          else {
40                  print "File '$_' does not exist.\n";
41          }
42  }
43
44  # If some of the files exists try to add ssh keys
45  if( $filename ) {
46          print "File '$filename' exists.\n";
47
48          # Check if running user has read access to the SSH key file
49          if( -r $filename ) {
50                  open FILE_R, "$filename" or die $!;
51
52                  # Check if provided SSH key has already been added
53                  while( <FILE_R> ) {
54                          # Remove end-of-line character from the line
55                          $_ =~ s/\n//;
56                          my $line = $_;
57
58                          for my $key ( @keys ) {
59                                  if( $line eq $key ) {
```

```
60                                      print "=== This SSH key has
    already been added to the file! ================\n";
61                                      print "'$filename'\n'$key'\n";
62                                      print
    "================================================================
    ==\n";
63                                      shift @keys; # Remove already
    added key from the array
64                                  }
65                          }
66                  }
67                  # Check if running user has write access to the SSH
    key file
68                  if( -w $filename ) {
69                          open FILE_W, ">> $filename" or die $!;
70
71                          # Write keys to the file
72                          for( @keys ) {
73                                  print FILE_W "$_\n";
74                          }
75
76                          # Did we add some keys or not?
77                          if( scalar @keys > 0 ) {
78                                  print "SSH key(s) added
    successfully!\n";
79                          }
80                          else {
81                                  print "All keys added already!\n";
82                          }
83                  }
84                  else {
85                          print "No write access to file
    '$filename'!\n";
86                  }
87          }
88          else {
89                  print "No read access to file '$filename'!\n";
90          }
91  }
92  else {
93          print "None of the files provided exists!\n";
94  }
95  exit;
```

LIITE 3: SE-AGENT-INSTALL.SH

```
1   #!/usr/bin/perl -w
2
3   use warnings;
4   use strict;
5
6   sub BEGIN{
7         push @INC, "/tmp/";
8   }
9   use SEAgent;
10
11  my %settings = (
12        server          => "http://10.8.129.80/se_agent2",
13        dlretry         => 3,
14        tmpdir          => "/tmp/cimex",
15        sniadir         => "/root/snia",
16        version         => "6.2.0-119",
17        dependency      => 1
18  );
19  my $installer = SEAgent->new();
20
21  my $force = 0;
22  if( @ARGV && $ARGV[0] eq "--force" ) {
23        print "Forcing reinstallation by using '--force'
    argument.\n";
24        $force = 1;
25  }
26
27  my $install_status = "";
28  if( !$force ) {
29        $install_status = $installer->checkIfInstalled();
30  }
31
32  if( $install_status eq "BOTH" ) {
33        if( $installer->testAgent($settings{tmpdir}) ) {
34              $installer->deleteInstallationFiles($settings{tmpdir},
    $settings{sniadir});
35              exit(0);
36        }
37        else {
38              $force = 1;
39        }
40  }
41
42  if( $force || $install_status eq "APPQcime" || $install_status eq
    "SNIA" ) {
43        $installer->deleteAgent($settings{sniadir},
    $settings{sniadir});
44  }
45
46  my @versions = ("","");
47  @versions = $installer->verifyVersions();
48
49  if( @versions ) {
50        my $os = $versions[0];
51        my $platform = $versions[1];
52
53        print "OS='$os'\nPLATFORM='$platform'\n";
54
55        if( $installer->copyInstallationFiles($settings{server}, $os,
    $platform, $settings{dlretry}, $settings{tmpdir},
    $settings{version}) ) {
56              if( $installer->installAgent($settings{tmpdir}, $os,
    $settings{server}, $settings{version}, $settings{dependency} ) ) {
57                    $installer->testAgent($os);
```

```
58                }
59        }
60        $installer->deleteInstallationFiles($settings{tmpdir},
    $settings{sniadir});
61  }
62  exit(0);
```

LIITE 4: SEAGENT.PM

```
1   package SEAgent;
2   use warnings;
3   use strict;
4
5   sub new {
6          my $self = {};
7          bless($self, "SEAgent");
8          return $self;
9   }
10
11  sub checkIfInstalled {
12         my $installed = "NONE";
13
14         # Check if APPQcime (SE agent) is installed
15         if ( qx(rpm -qa APPQcime) ) {
16                print "APPQcime installed.\n";
17                $installed = "APPQcime";
18         }
19         else {
20                print "APPQcime not installed.\n";
21         }
22         if( -e "/etc/hba.conf" && -e "/usr/lib/libqlsdm.so" ) {
23                print "SNIA API installed.\n";
24
25                if( $installed eq "APPQcime" ) {
26                       $installed = "BOTH";
27                }
28                else {
29                       $installed = "SNIA";
30                }
31         }
32         else {
33                print "SNIA API not installed.\n";
34         }
35         return $installed;
36  }
37
38  sub verifyVersions {
39         my $os = "";
40         if( qx(rpm -qa *release) =~ /release-5/ ) {
41                $os = "rhel5";
42         }
43         elsif( qx(rpm -qa *release) =~ /release-4/ ) {
44                $os = "rhel4";
45         }
46         elsif( qx(rpm -qa *release) =~ /release-3/ ) {
47                $os = "rhel3";
48         }
49         else {
50                print "FAILED to verify operating system!
    Exiting...\n";
51                return 0;
52         }
53         my $platform = qx(uname -i); chomp($platform);
54         if( $platform ne "x86_64" && $platform ne "i386" ) {
55                print "Failed to verify operating system's platfrom
    (x86_64 or i386)! Exiting...\n";
56                return 0;
57         }
58         return ($os, $platform);
59  }
60
61  sub copyInstallationFiles {
62         # Subrutine parameters:
```

```
63          my $server = $_[1];
64          my $os = $_[2];
65          my $platform = $_[3];
66          my $dlretry = $_[4];
67          my $tmpdir = $_[5];
68          my $version = $_[6];
69
70          my $output = 1;
71          my $retries = $dlretry;
72
73          while( $retries > 0 && $output != 0 ) {
74                  $output = system("wget -q -P$tmpdir/qlapi
    $server/qlapi/qlapi.tgz");
75                  if( $output != 0 ) {
76                          print "Downloading SNIA API files FAILED!
    Retrying... (retry times left: $retries)\n";
77                          sleep(10);
78                          --$retries;
79                  }
80                  else {
81                          print "SNIA HBA API copied succesfully from:
    $server/\n";
82                  }
83          }
84          if( $output == 0 ) {
85                  $output = 1;
86                  $retries = $dlretry;
87                  while( $retries > 0 && $output != 0 ) {
88                          $output = system("wget -q -P$tmpdir
    $server/$os/$platform/APPQcime-Requires-$version.$platform.rpm");
89                          if( $output != 0 ) {
90                                  print "Downloading APPQcime-Requires
    FAILED! Retrying... (retry times left: $retries)\n";
91                                  sleep(10);
92                                  --$retries;
93                          }
94                  }
95                  if( $output == 0 ) {
96                          print "APPQcime-Requires copied succesfully
    from: $server\n";
97                          $output = 1;
98                  }
99                  while( $retries > 0 && $output != 0 ) {
100                         $output = system("wget -q -P$tmpdir
    $server/APPQcime-$version-i386.rpm");
101                         if( $output != 0 ) {
102                                 print "Downloading APPQcime file
    FAILED! Retrying... (retry times left: $retries)\n";
103                                 sleep(10);
104                                 --$retries;
105                         }
106                 }
107                 if( $output == 0 ) {
108                         print "APPQcime copied succesfully from:
    $server\n";
109                         return 1;
110                 }
111         }
112         print "FAILED to download installation files from: $server
    !\n";
113         return 0;
114 }
115
116 sub installAgent {
117         my $tmpdir = $_[1];
118         my $os = $_[2];
119         my $server = $_[3];
120         my $version = $_[4];
121         my $dependency = $_[5];
122
123         my $output = system("cd $tmpdir/qlapi; tar -xzf *.tgz");
124         if( $output == 0 ) {
125                 if( $os ne "rhel3" ) {
126                         print "Installing SNIA API...\n";
127                         system("cd $tmpdir/qlapi; ./libinstall");
128                         if( $os eq "rhel4" ) {
```

```
129                             qx(modprobe qioctlmod);
130                             print "Module found (rhel4)?", qx(lsmod
    | grep qioctlmod);
131                     }
132             }
133             if( $dependency ) {
134                     if( !qx(rpm -qa compat-libstdc++-296) ) {
135                             print "Package compat-libstdc++-296 not
    installed. Trying to download...\n";
136                             $output = system("wget -q -P$tmpdir
    $server/compat-libstdc++-296.i386.rpm");
137                             if( $output != 0 ) {
138                                     print "Downloading compat-
    libstdc++-296.i386 FAILED! Exiting...\n";
139                                     return 0;
140                             }
141                             else {
142                                     print "Downloading compat-
    libstdc++-296.i386 completed!\n";
143                                     sleep(5);
144                             }
145                             print "Installing compat-libstdc++-
    296.i386...\n";
146                             $output = system("cd $tmpdir/; rpm -idh
    compat-libstdc++-296.i386.rpm");
147
148                             if( $output == 0 ) {
149                                     print "Package compat-
    libstdc++-296.i386 installed successfully\n";
150                             }
151                             else {
152                                     print "Installation of compat-
    libstdc++-296.i386 FAILED! Exiting...\n";
153                                     return 0;
154                             }
155                     }
156             }
157             $output = system("cd $tmpdir/; rpm -idh APPQcime-
    Requires-$version*.rpm");
158
159             print "Installing APPQcime...\n";
160             $output = system("cd $tmpdir/; rpm -idh APPQcime-
    $version-i386.rpm");
161             if( $output == 0 ) {
162                     print "APPQcime installed successfully.\n";
163             }
164             else {
165                     print "APPQcime installation FAILED!
    Exiting...\n";
166                     return 0;
167             }
168       }
169       return 1;
170 }
171
172 sub testAgent {
173
174       if( qx(/opt/APPQcime/tools/status) =~ /not running/ ) {
175             # Start the agent if it is not running
176             print qx(/opt/APPQcime/tools/start);
177       }
178
179       if( qx(/opt/APPQcime/tools/status) =~ /Process id: [0-9]+/ )
    {
180             my @output = qx(/opt/APPQcime/tools/hbatest -v);
181             for(@output) {
182                     if( $_ =~ /Number of HBA's is 0/ ) {
183                             print "No HBAs detected in the
    system!\n";
184                     }
185                     elsif( $_ =~ /Number of HBA's is ([1-9]{1})/ )
    {
186                             print "Found $1 HBA adapters.\n";
187                     }
188                     elsif( $_ =~ /Adapter number [0-9]{1} is
    named:/ ) {
```

```
189                             print "$_";
190                     }
191                     elsif( $_ =~ /PortIndex:[0-9]{1}/ ) {
192                             print "$_";
193                     }
194                     elsif( $_ =~ /Target Mappings: ([1-9]+)/ ) {
195                             print "  $&\n";
196                     }
197                     elsif( $_ =~ /scsiID:/ ) {
198                             print "  $_";
199                     }
200                     elsif( $_ =~ /FcpId:/ ) {
201                             print "    $_";
202                     }
203             }
204             return 1;
205     }
206     else {
207             print "SE Agent not running and FAILED to start!";
208     }
209     return 0;
210 }
211
212 sub deleteInstallationFiles {
213     #Parameters:
214     my $tmpdir = $_[1];
215     my $sniadir = $_[2];
216     # Move the SNIA API uninstaller to a different location so it
    can be used later if necessary
217     if( -e "$tmpdir/qlapi/libremove" ) {
218             qx(mkdir $sniadir/; mv $tmpdir/qlapi/libremove
    $sniadir/);
219     }
220     if( -e $tmpdir ) {
221             qx(rm -rf $tmpdir);
222     }
223 }
224
225 sub deleteAgent {
226     my $sniadir = $_[1];
227
228     my $output = system("$sniadir/libremove");
229     if( $output == 0 ) {
230             # Delete previous uninstaller
231             qx(rm -rf $sniadir/);
232     }
233     else {
234             print "Deleting SNIA API FAILED!\n";
235     }
236     if( qx(rpm -qa APPQcime-Requires) ) {
237             system("rpm -e APPQcime-Requires");
238     }
239     if( qx(rpm -qa APPQcime) ) {
240             print "Deleting APPQcime...\n";
241             $output = system("rpm -e APPQcime");
242             if( $output == 0 ) {
243                     print "APPQcime deleted successfully.\n";
244                     return 1;
245             }
246             else {
247                     print "Deleting APPQcime FAILED!\n";
248             }
249     }
250     return 0;
251 }
252 1;
```

LIITE 5: ASENNUSSKRIPTIN ESIMERKKITULOSTE

```
1   Forcing reinstallation by using '--force' argument.
2   Removing QLogic HBA API library...
3   Done.
4   OS='rhel4'
5   PLATFORM='x86_64'
6   APPQcime-Requires copied succesfully from: http://<PALVELIN>/se_agent
7   APPQcime copied succesfully from: http://<PALVELIN>/se_agent
8   Installing SNIA API...
9   Setting up QLogic HBA API library...
10  Please make sure the /usr/lib/libqlsdm.so file is not in use.
11  Installing 32bit api binary for x86_64.
12  Installing 64bit api binary for x86_64.
13  Done.
14  Module found (rhel4)?qioctlmod              71361  0
15  qla2xxx               196385  18 qioctlmod,qla2400
16  Package compat-libstdc++-296 not installed. Trying to download...
17  Downloading compat-libstdc++-296.i386 completed!
18  Installing compat-libstdc++-296.i386...
19  ##################################################
20  ##################################################
21  Package compat-libstdc++-296.i386 installed successfully
22  Installing APPQcime...
23  ##################################################
24  ##################################################
25  APPQcime installed successfully.
26  Starting CIM Extension for LINUX...
27  Found 2 HBA adapters.
28  Adapter number 0 is named: qlogic-qla2xxx-0
29    PortIndex:0
30    Target Mappings: 4
31    scsiID:/dev/sda;/dev/sg0;:c0t0s1
32      FcpId: ID:990f00, NWWN:0x50060E8015260D21 PWWN:0x50060E8015260D21
    FcpLun:0x100
33      FcpId: ID:990f00, NWWN:0x50060E8015260D21 PWWN:0x50060E8015260D21
    FcpLun:0x100
34    scsiID:/dev/sdb;/dev/sg1;:c0t0s2
35      FcpId: ID:990f00, NWWN:0x50060E8015260D21 PWWN:0x50060E8015260D21
    FcpLun:0x200
36      FcpId: ID:990f00, NWWN:0x50060E8015260D21 PWWN:0x50060E8015260D21
    FcpLun:0x200
37    scsiID:/dev/sdc;/dev/sg2;:c0t0s3
38      FcpId: ID:990f00, NWWN:0x50060E8015260D21 PWWN:0x50060E8015260D21
    FcpLun:0x300
39      FcpId: ID:990f00, NWWN:0x50060E8015260D21 PWWN:0x50060E8015260D21
    FcpLun:0x300
40    scsiID:/dev/sdd;/dev/sg3;:c0t0s4
41      FcpId: ID:990f00, NWWN:0x50060E8015260D21 PWWN:0x50060E8015260D21
    FcpLun:0x400
42      FcpId: ID:990f00, NWWN:0x50060E8015260D21 PWWN:0x50060E8015260D21
    FcpLun:0x400
43  Adapter number 1 is named: qlogic-qla2xxx-1
44    PortIndex:0
```